The Macroeconomic Effects of Fiscal Adjusments in The UK

Samid Ali

Contents

1 Abstract

2 Introduction

3 Literature Review

3.1 Costs of high indebtedness

Sutherland, Hoeller, and Merola [3] draw attention to the fiscal challenges facing countries following the Global Financial Crisis, noting that gross government debt has exceeded 100% of GDP for the OECD as an aggregate. This has been exacerbated following the Covid pandemic where governments implemented fiscal measures to mitigate the economic costs of the pandemic. Makin and Layton (2021) highlight that governments must employ fiscal responsibility to protect their economies from the risks that high indebtedness exposes them to. Warmedinger, Checherita-Westphal, and De Cos [4] emphasise the importance of public debt sustainability for ensuring macroeconomic stability. There are several mechanisms through which excessive debt could harm the economy. For instance, concerns regarding public finances could reduce business confidence, leading to decreased investment and thus a slowdown in growth. Additionally, strained public finances could hinder the ability of economies to react counter cyclically to economic shocks. This rationale is supported by the IMF (2023) who argue that economies should rebuild their fiscal buffers to reduce their debt vulnerabilities. Therefore, fiscal consolidation is clearly needed to ensure the long-term resilience of the economy. As a target, Sutherland, Hoeller, and Merola [3] propose that countries should aim to bring debt levels towards 50% of GDP: a figure which would require the UK to halve its current debt levels (ONS, 2025). The IMF (2023) argues that to stabilise GDP, fiscal adjustments should be in the region of up to 4% of GDP. Thus, achieving this objective would require significant fiscal adjustments, motivating further research to support the transition towards more sustainable public finances. Alesina, Favero, and Giavazzi (2012) find that permanent fiscal adjustments have lower output costs, interpreting this result as due to business confidence. (easier to forecast when fiscal adjustments are more predictable? Thus have less of a effect on confidence)

Kumar and Woo (2015) find that greater indebtedness is associated with lower economic growth. They also find noticeable nonlinearities in this result, with the most severe effect when public indebtedness exceeds 90% of GDP. Given the aforementioned level of public indebtedness in the UK ...
Blanchard (2019) argue that even when the interest rate is less than the growth rate, and thus there is no fiscal cost of high indebtedness, there may still be a welfare cost due to reduced capital accumulation.

3.2 Fiscal Consolidaton

While the importance of fiscal consolidation has been highlighted, it is crucial that these measures are not at the expense of the broader economy. By investigating forecast errors for a sample of European countries, Blanchard and Leigh [1] find that larger anticipated fiscal consolidation was associated with lower growth. This result was interpreted as due to the fiscal multiplier being greater than anticipated by forecasters. Consequently, fiscal tightening would have further dampened demand, and thus improvements in fiscal consolidation would be offset by reduced growth. Gechert (2019) adopt a similar methodology, finding that austerity measures in the Euro Area deepened the crisis, contributing to hysterisis effects. Fatas and Summers (2018) extend this research, investigating the long-term effects that fiscal adjustments have had on GDP. Their analysis suggests that fiscal consolidations have failed to lower the debt-to-GDP ratio due to a hysteresis effect of contractionary fiscal policy. This research underscores the need for effectively quantifying fiscal multipliers to understand potential trade-offs between various economic objectives. Ilzetzki, Mendoza, and Végh [2] provide further insight into the fiscal multiplier, suggesting that the heterogeneity in the estimates reported in the literature can be attributed to differences in structural characteristics of the economy considered. This reinforces the importance of research to better understand the fiscal multiplier for different policy instruments, particularly as this may vary across countries and over time. Additionally, Alesina et al (2015) compare multipliers due to spending and tax adjustments. They find that … have more severe effects, attributing this to reduced business confidence. Alesina et al (2002) investigate the effect of fiscal policy on investment.

3.3 Synthesis of Methodology

Capek and Cuasera (2020) simulate 20 million fiscal multipliers, highlighting how methodological choices contribute to the heterogeneity in estimates of fiscal multipliers prevalent in the literature. Consequently, they advocate for explicitly outlining modelling choices and assumptions. Similarly, Gechert (2017) provides a synthesis of the methodologies used to estimate fiscal multipliers, highlighting competing definitions for the fiscal multiplier and possible issues in its estimation. Among these issues, Gechert (2017) highlight potential omitted variable bias in the VAR model (motivating the use of additional controls such as the price level and real interest rate), anticipation effects (cf Leepper + Zha fiscal foresight), and nonlinearities.

Structural Vector Autoregressions (SVARs) have been prominent in the literature to estimate fiscal multipliers. Various approaches to identification have been used, with XXX (YYYY) noting that after accounting for the empirical specification, the competing identifying approaches have little effect on the estimated multipliers. Blanchard and Perotti (2002) pioneered this strand of the research, leveraging methodologies previously popularised by the monetary economics. To identify their SVAR, Blanchard and Perotti leverage instituional information. They provide a definition for the fiscal variables and highlight that government expenditure is predetermined within a quarter. Recursive measures to identfication have been employed by Fatas and Mihov (YYYY) and Fernandex (2008). Fernandez argues that Uhlig and Mountford (200Y) apply restrictions on the signs of the impulse response functions. Caldara and Kamps (2008) reviews the literature on SVAR identification. Caldara and Kamps (2017) introduce a new approach for identification.

Add DSGE lit for context

3.4 Data Proc above

4 Econometric Methodology

This section outlines the VAR methodology that will be employed by this research. The reduced-form VAR model can be written as:

$$X_t = \mu_0 + \mu_1 + A_1 X_{t-1} + A_2 X_{t-2} + \cdots + X_p Y_{t-p} + \epsilon_t$$

where ϵ_t is the vector of reduced-form residuals and p determines the lag length. In line with XXX we assume the model contains 4 lags. Given the use of quarterly data, this can be interpreted as lags of the model variables having a direct affect for up to a year. This use of lag is supported by the Akaike Information Criteria. The VAR model above assumes that the data generating process includes a deterministic linear time trend and constant intercept. This is included to mitigate for potential spurious regression between trending factors in the endogenous variables and to account for their nonzero mean.

Regarding the endogenous variables considered, Blanchard and Perotti (2002) investigate the effects of fiscal shocks using a three-dimensional vector autoregressive model consisting of GDP, government expenditure, and governemnet revenue. While such a model could be used to estimate the effects of fiscal shocks, Gechert (2017) highlights the potential issues of omitted variable bias. Consequently we augment the model to include also a short term interest rate, the GDP deflator rate, and UK exports. These variables are included to account for the effects of monetary policy, price levels, and trade respectively. Consequently, the impulse response functions reported later are better interpreted as the response of GDP to the fiscal variables.

The data is used at a quarterly frequency from 1987:1 to 2023:3. As well as allowing for more precise estimates, this level of frequency is necessary for the identification of structural shocks, as outlined in the following section. The fiscal variables are defined at the general level and follow the European System of Accounts (ESA, 2010). In particular, government expenditure represents the outflows associated with government activities, including consumption, investment, and transfers. The inflows to the government, government revenue, consists of receipts net of transfer and interest payments. Cf Blanchard and Perotti (2002).

Following Fernandez (2006), the natural logarithm of these variables is used, with the exception of the short term interest rate, which enters the model in levels. Furthermore, the fiscal variables and GDP are used in real terms. Capek and Cuaresma (2020) highlight that data used to estimate fiscal multipliers is typically seasonally adjusted. Therefore (with the exception of GDP which was sourced after seasonal adjustment) variables have been seasonally adjusted using X-13ARIMA-SEATS method. The intention of this is tease out the noise from the data so that the model can estimate the underlying trends.

This research is interested in assessing whether fiscal multipliers have stayed consistent following the Global Financial Crisis. Consequently we will 3 specifications, firstly the restricted model, pooling data from the entire period. We also run 2 separate models, before and after the GFC to account for possible changes in the relationship between the fiscal variables and GDP following the crisis. This will allow us to formally test the stability of parameters over the period by using a Chow test.

- debt sustainability

$$X_t = \begin{pmatrix} G_t \\ R_t \\ GDP_t \\ \tau_t \\ P_t \end{pmatrix}$$

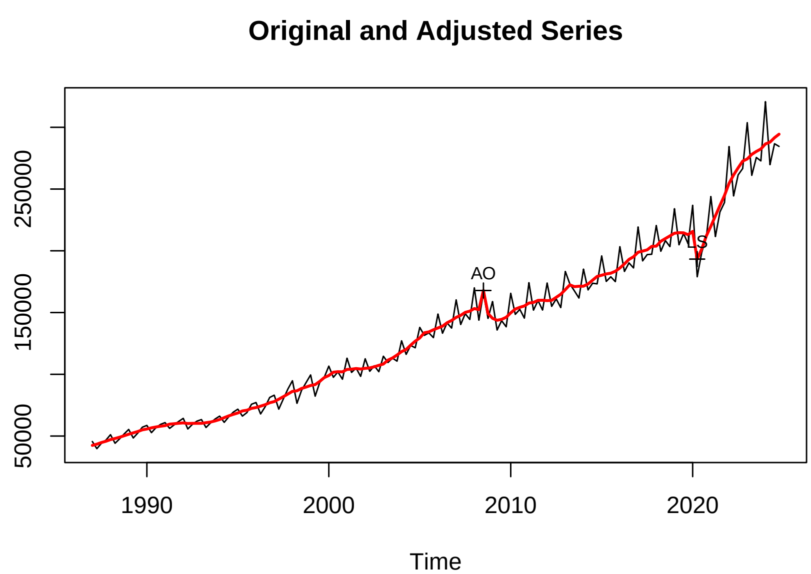
Discuss Empirical model, lag structure, data sources, potentially features of the data

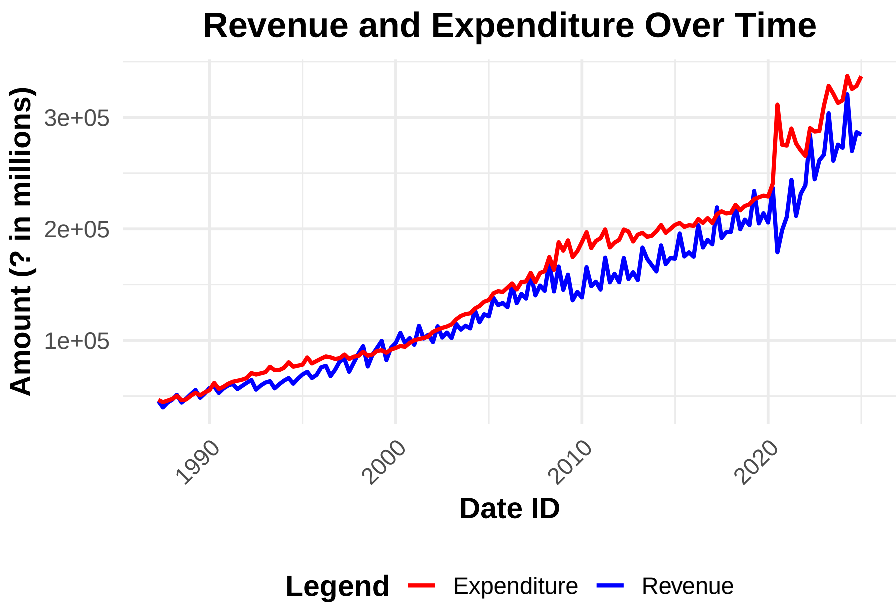
4.1 Data Sources

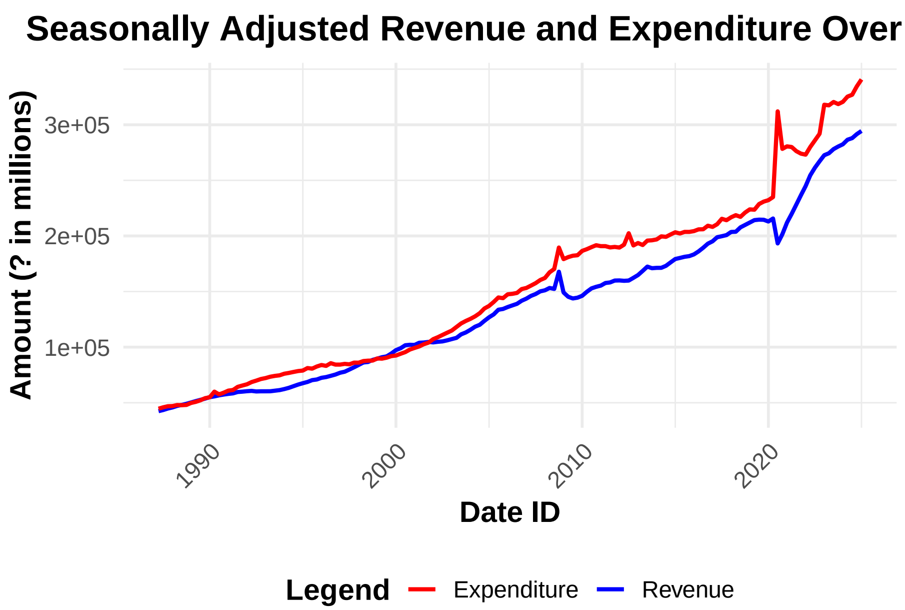
- Fiscal Variables (not seasonally adjusted): https://www.ons.gov.uk/economy/governmentpublicsectorandtaxes/publicspending/datasets/esatable25quarterlynonfinancialaccountsofgeneralgovernment
- UK Exports (seasonally Adjusted, %): https://fred.stlouisfed.org/series/XTEXVA01GBQ188S
- UK Exports (seasonally Adjusted, £millions): https://fred.stlouisfed.org/series/NXRSAXDCGBQ

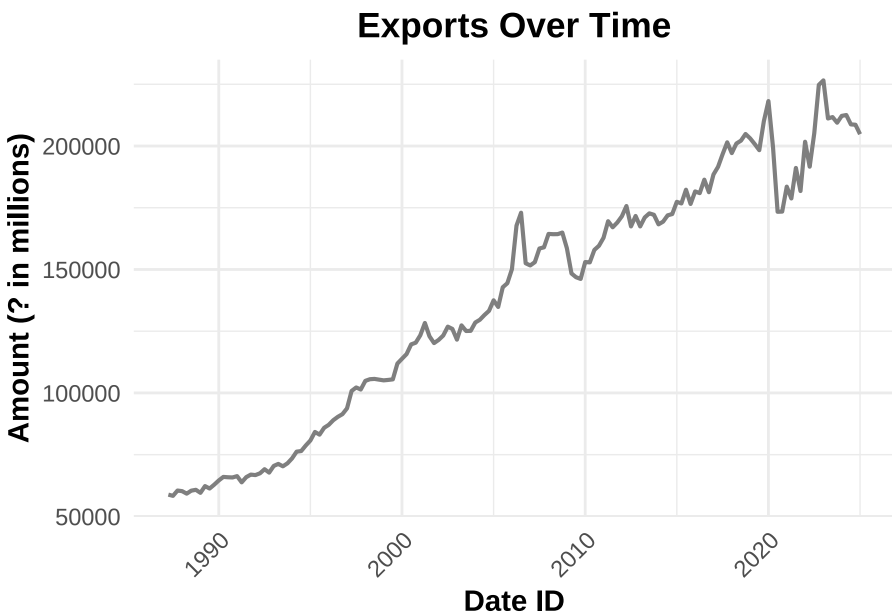
- LFS (Pop aged 16-64):
https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/employmentandemployeetypes/timeseries/lf2o/lms
- GDP (SA) and defaltor rate:
https://www.ons.gov.uk/economy/grossdomesticproductgdp/bulletins/quarterlynationalaccounts/latest#data-on-gdp-quarterly-national-accounts
- interest rate (SR. This is dates of changes to the policy rate. Have interpolated to get quarterly data):
https://www.bankofengland.co.uk/monetary-policy/the-interest-rate-bank-rate
- 3 month interest rate:
https://fred.stlouisfed.org/series/IR3TIB01GBM156N

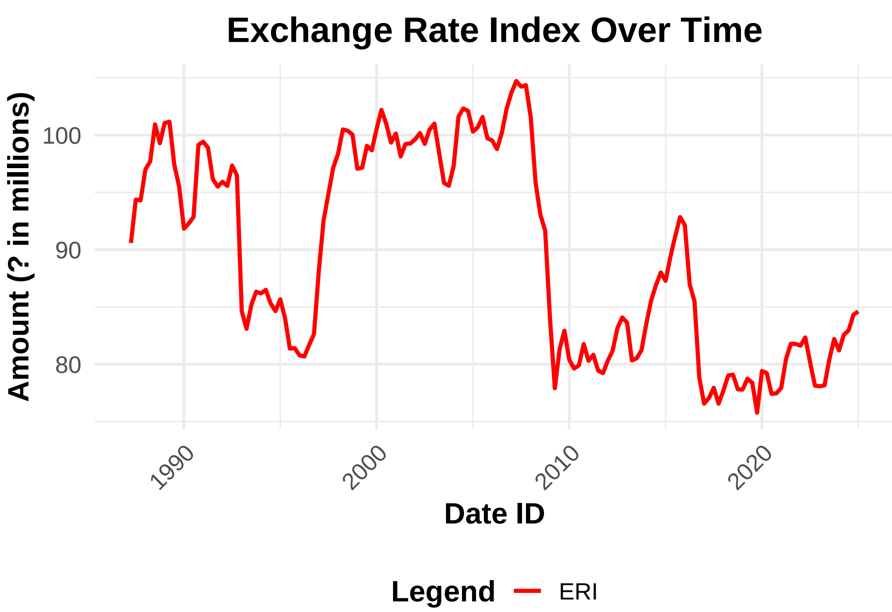
4.2 Plots

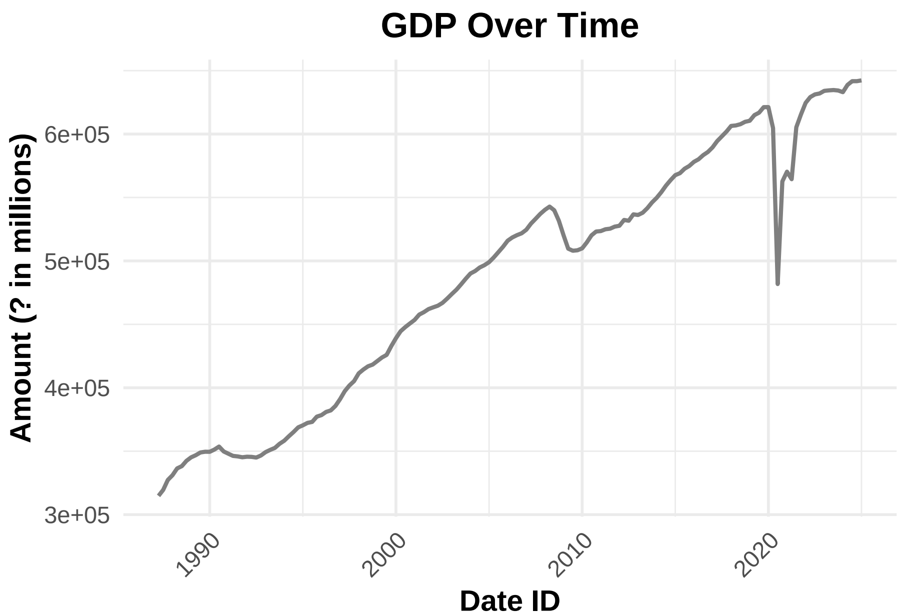


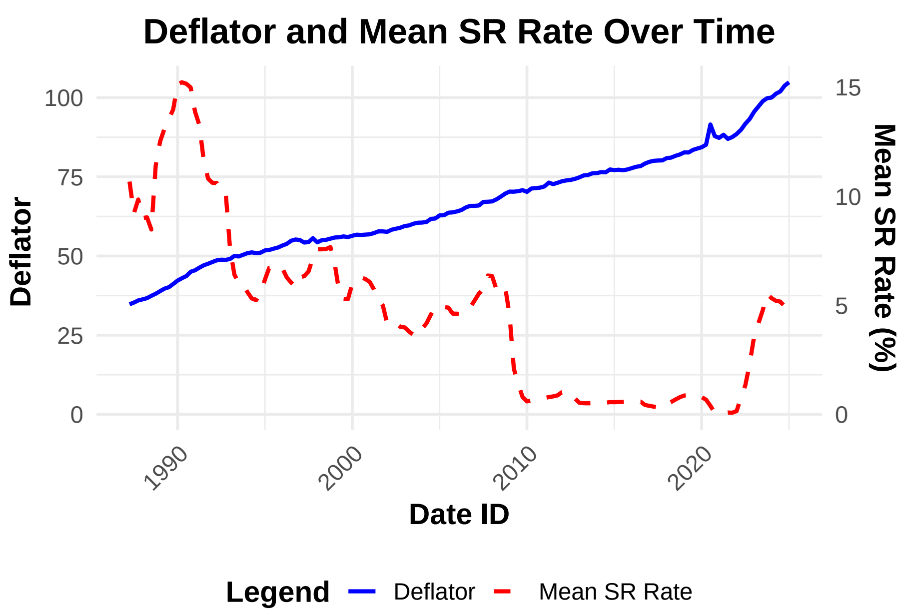

Revenue and Expenditure Over Time
Amount (? in millions)
3e+05
2e+05
1e+05
1990
2000
2010
2020
Date ID
Legend
Expenditure
Revenue

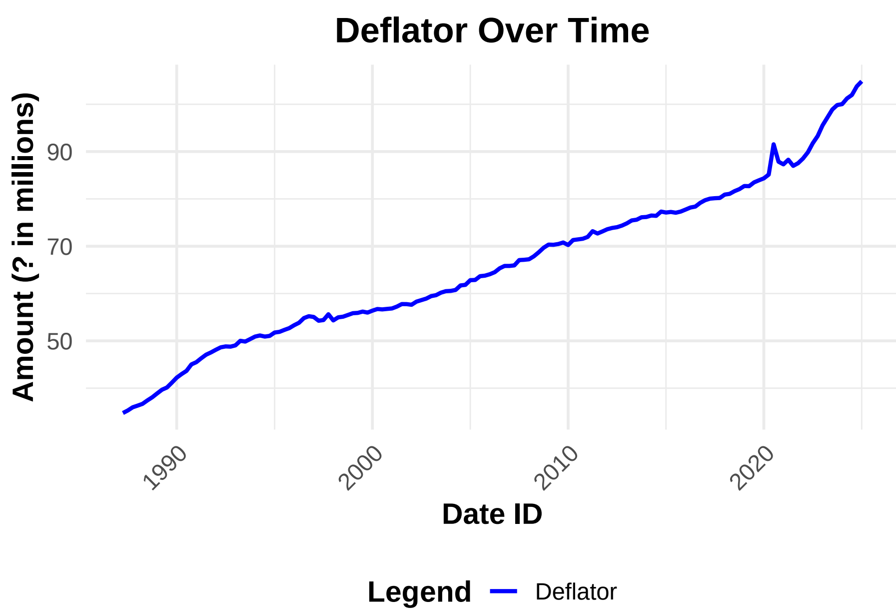

Seasonally Adjusted Revenue and Expenditure Over
Amount (? in millions)
3e+05
2e+05
1e+05
1990
2000
2010
2020
Date ID
Legend
Expenditure
Revenue


Exports Over Time
Amount (? in millions)
200000
150000
100000
50000
1990
2000
2010
2020
Date ID


Exchange Rate Index Over Time
Amount (? in millions)
100
90
80
1990
2000
2010
2020
Date ID
Legend
ERI


GDP Over Time
Amount (? in millions)
6e+05
5e+05
4e+05
3e+05
1990
2000
2010
2020
Date ID


Deflator and Mean SR Rate Over Time
Deflator
100
75
50
25
0
Mean SR Rate (%)
15
10
5
0
1990
2000
2010
2020
Date ID
Legend
Deflator
Mean SR Rate



4.3 Model

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      2      2      1      2
```

```
## [1] 0.4270143 0.3757661 0.3476387 0.3476387 0.3319190 0.1456201
```

```
##  [1] 0.9533752 0.9533752 0.9420585 0.8545320 0.8545320 0.7847829 0.7847829
##  [8] 0.6959972 0.6959972 0.6839151 0.6839151 0.6613079 0.6613079 0.6031976
## [15] 0.5905326 0.5905326 0.5030031 0.5030031 0.4595746 0.4595746 0.4279766
## [22] 0.3730242 0.2866050 0.2866050
```

Note: VAR analysis requires stability of the system. Need to find code to ensure the eigenvalues of the autoregressive roots lie within the unit circle.

5 Identification

Explaining distinction between the reduced form and structural model. Interested in the structural shocks which we will recover as follows.

Let X_t be the vector of variables:

$$X_t = \begin{pmatrix} G_t \\ R_t \\ GDP_t \\ T_t \\ P_t \end{pmatrix}$$

The reduced-form VAR model can be written as:

$$X_t = A_1 X_{t-1} + A_2 X_{t-2} + \cdots + A_p X_{t-p} + \epsilon_t$$

where ϵ_t is the vector of reduced-form residuals. To recover the structural shocks u_t, we assume:

$$\epsilon_t = B u_t$$

$B^{-}1$ is the structural impact multiplier matrix.

where B is a lower triangular matrix. The structural shocks u_t are assumed to be uncorrelated and have unit variance.

The matrix B can be obtained using Cholesky decomposition of the covariance matrix of the reduced-form residuals Σ_ϵ:

$$\Sigma_\epsilon = E[\epsilon_t \epsilon_t'] = BB'$$

Given the recursive ordering (G, R, GDP, T, P), the matrix B has the form:

$$B = \begin{pmatrix} b_{11} & 0 & 0 & 0 & 0 \\ b_{21} & b_{22} & 0 & 0 & 0 \\ b_{31} & b_{32} & b_{33} & 0 & 0 \\ b_{41} & b_{42} & b_{43} & b_{44} & 0 \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} \end{pmatrix}$$

Thus, the structural shocks u_t can be recovered as:

$$u_t = B^{-1} \epsilon_t$$

Killian and Lutkepohl (2017) highlight that identification of the structural parameters is not a purely statistical concern, the restrictions must also be economically meaningful for the resulting structural parameters to be identified. Therefore we proceed with an exposition of the economic assumptions implicit in the impact multiplier matrix, B, comparable to Fernandez (2006).

1) Blanchard and Perotti (2002) argue that the use of quarterly data allows government spending to be interpreted as predetermined with respect to the rest of the variables within the quarter. This is motivated by implementation lags for changes to government spending and consequently this is ordered first.

2) Given physical constraints, the interest rate is assumed not to react contemporaneously to price, net taxes, output, or exports. Thus the short term rate is considered the next most exogenous variable.

3) However monetary policy shocks are assumed to affect output, net taxes, and prices contemporaneously. Fernandez (2006) justifies this assumption by noting that interest movements are anticipated and thus they can be transmitted to real variables relatively quickly.
   NB on the appropriateness of the assumption that interest rate does not react to price/ output.

3a) By construction exports contemporaneously affect GDP and revenue. Additionally exports affect the price level through

4) Due to price stickiness, prices do not react contemporaneously to shocks to GDP,

5) Due to physical constraints in adjusting consumption and investment, net taxes are assumed not to affect economic activity.

6 Results

6.1 Unrestricted Model

6.1.1 SVAR Recode

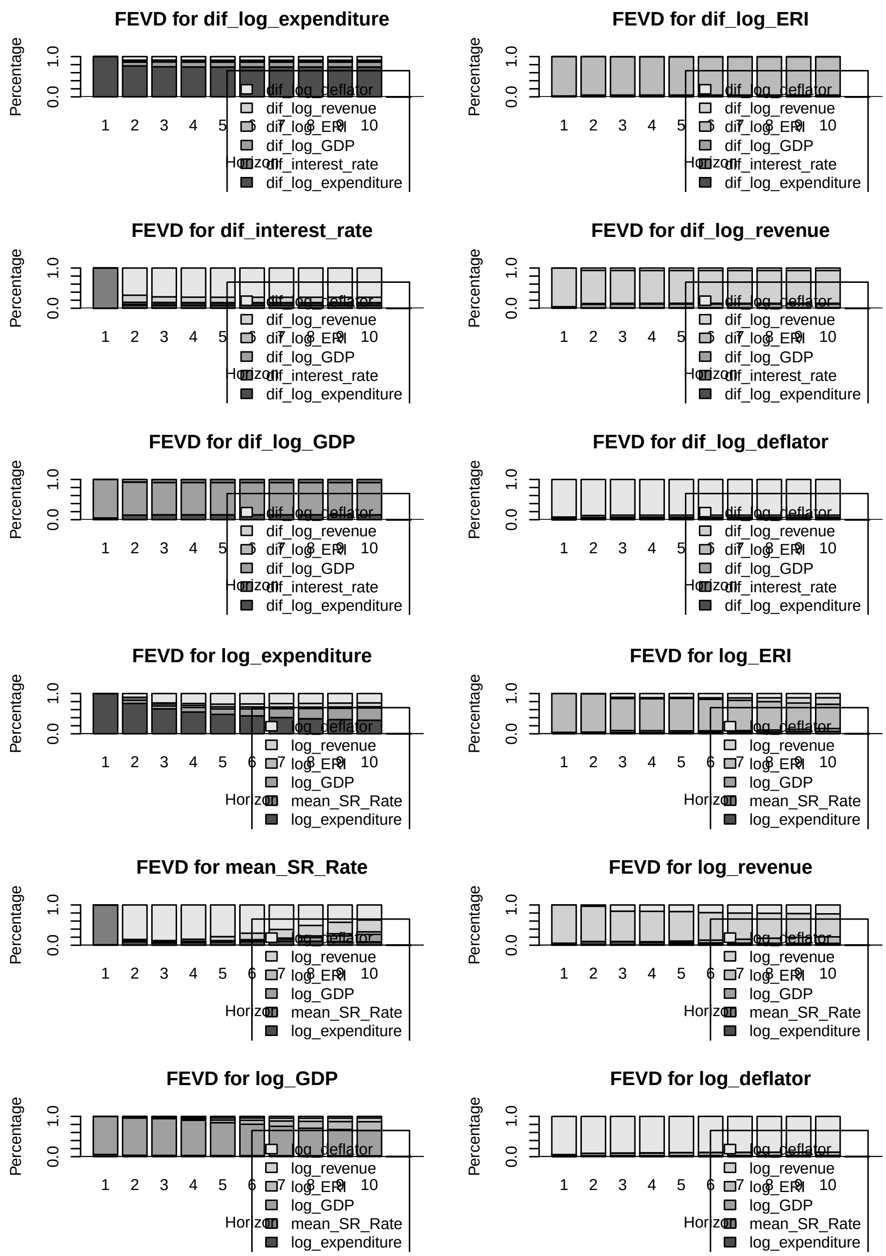
```
##                      dif_log_expenditure dif_interest_rate  dif_log_GDP
## dif_log_expenditure           0.02877684      -0.006372266 -0.017138164
## dif_interest_rate             0.00000000       0.499234158  0.003117498
## dif_log_GDP                   0.00000000       0.000000000  0.014776701
## dif_log_ERI                   0.00000000       0.000000000  0.000000000
## dif_log_revenue               0.00000000       0.000000000  0.000000000
## dif_log_deflator              0.00000000       0.000000000  0.000000000
##                       dif_log_ERI dif_log_revenue dif_log_deflator
## dif_log_expenditure -0.002070251   -0.0051139311     5.897077e-03
## dif_interest_rate    0.007742278    0.0057165176     7.772622e-04
## dif_log_GDP          0.002276047    0.0118243692    -3.506140e-03
## dif_log_ERI          0.023536066    0.0003831962    -4.128475e-05
## dif_log_revenue      0.000000000    0.0129288049     1.236249e-03
## dif_log_deflator     0.000000000    0.0000000000     6.736031e-03
```

```
##                     dif_log_expenditure dif_interest_rate dif_log_GDP
## dif_log_expenditure          1.00000000         0.0000000  0.00000000
## dif_interest_rate           -0.01276408         1.0000000  0.00000000
## dif_log_GDP                 -1.15980986         0.2109738  1.00000000
## dif_log_ERI                 -0.08796078         0.3289538  0.09670466
## dif_log_revenue             -0.39554554         0.4421536  0.91457557
## dif_log_deflator             0.87545273         0.1153887 -0.52050525
##                      dif_log_ERI dif_log_revenue dif_log_deflator
## dif_log_expenditure  0.000000000       0.0000000                0
## dif_interest_rate    0.000000000       0.0000000                0
## dif_log_GDP          0.000000000       0.0000000                0
## dif_log_ERI          1.000000000       0.0000000                0
## dif_log_revenue      0.029638948       1.0000000                0
## dif_log_deflator    -0.006128943       0.1835277                1
```

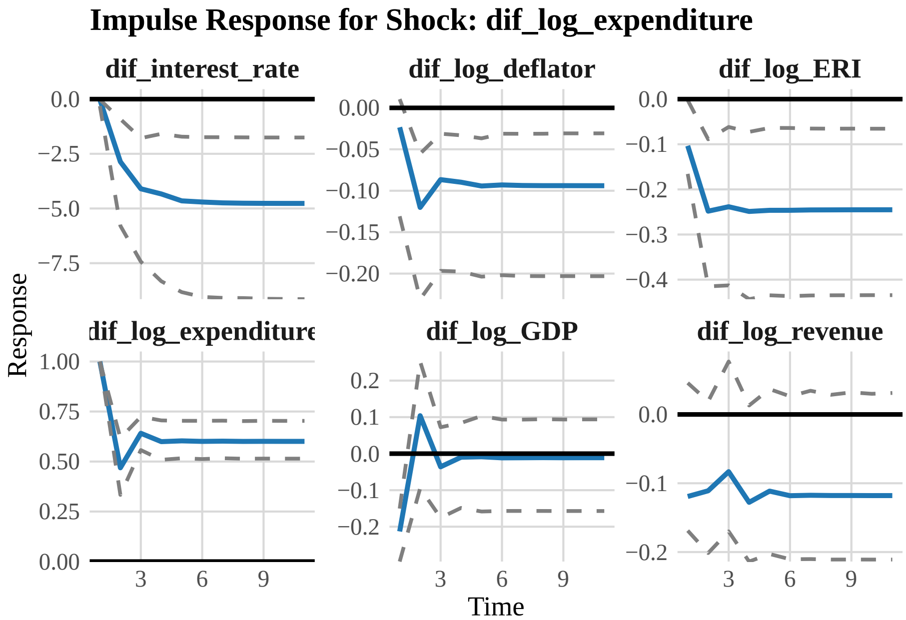
6.1.1.1 IRFs recode

6.1.2 SVAR functions

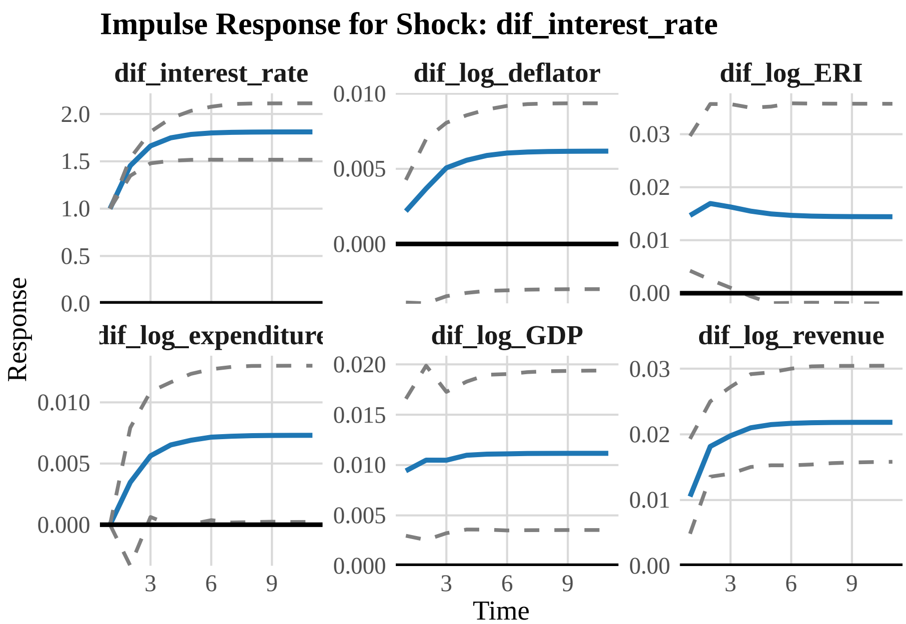
```
##
## SVAR Estimation Results:
## ========================
##
##
## Estimated A matrix:
##                     dif_log_expenditure dif_interest_rate dif_log_GDP
## dif_log_expenditure             1.00000          0.000000     0.00000
## dif_interest_rate               0.02777          1.000000     0.00000
## dif_log_GDP                     0.21278         -0.009425     1.00000
## dif_log_ERI                     0.12673         -0.015722     0.11155
## dif_log_revenue                 0.15352         -0.013551     0.09133
## dif_log_deflator                0.09992         -0.008014     0.18732
##                     dif_log_ERI dif_log_revenue dif_log_deflator
## dif_log_expenditure      0.0000          0.0000                0
## dif_interest_rate        0.0000          0.0000                0
## dif_log_GDP              0.0000          0.0000                0
## dif_log_ERI              1.0000          0.0000                0
## dif_log_revenue          0.1470          1.0000                0
## dif_log_deflator         0.1467          0.1815                1
```


FEVD for dif_log_expenditure
FEVD for dif_log_ERI
FEVD for dif_interest_rate
FEVD for dif_log_revenue
FEVD for dif_log_GDP
FEVD for dif_log_deflator
FEVD for log_expenditure
FEVD for log_ERI
FEVD for mean_SR_Rate
FEVD for log_revenue
FEVD for log_GDP
FEVD for log_deflator
Percentage
0.0
1.0
1 2 3 4 5 6 7 8 9 10
Horizon
dif_log_deflator
dif_log_revenue
dif_log_ERI
dif_log_GDP
dif_interest_rate
dif_log_expenditure
log_deflator
log_revenue
log_ERI
log_GDP
mean_SR_Rate
log_expenditure

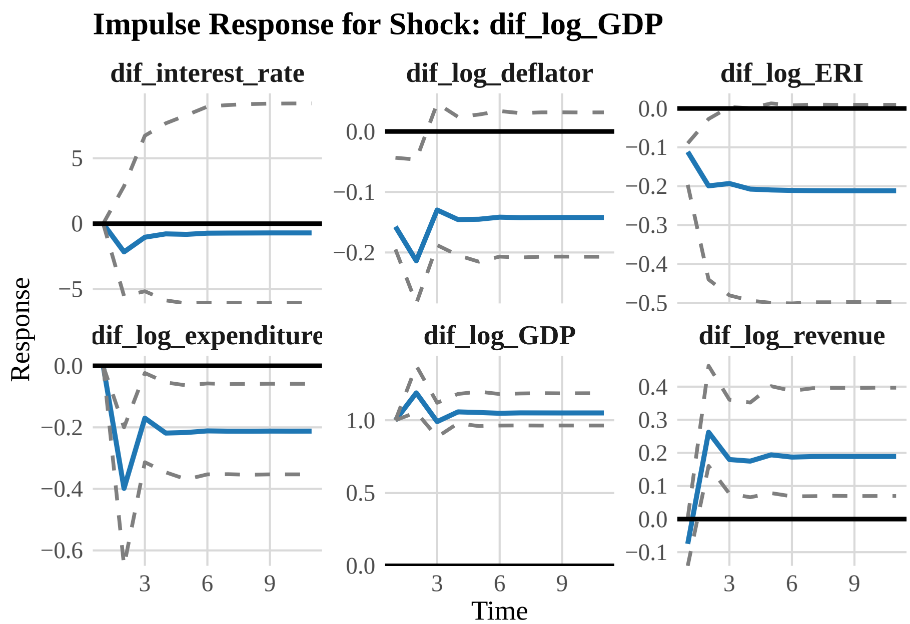
6.2 IRFs

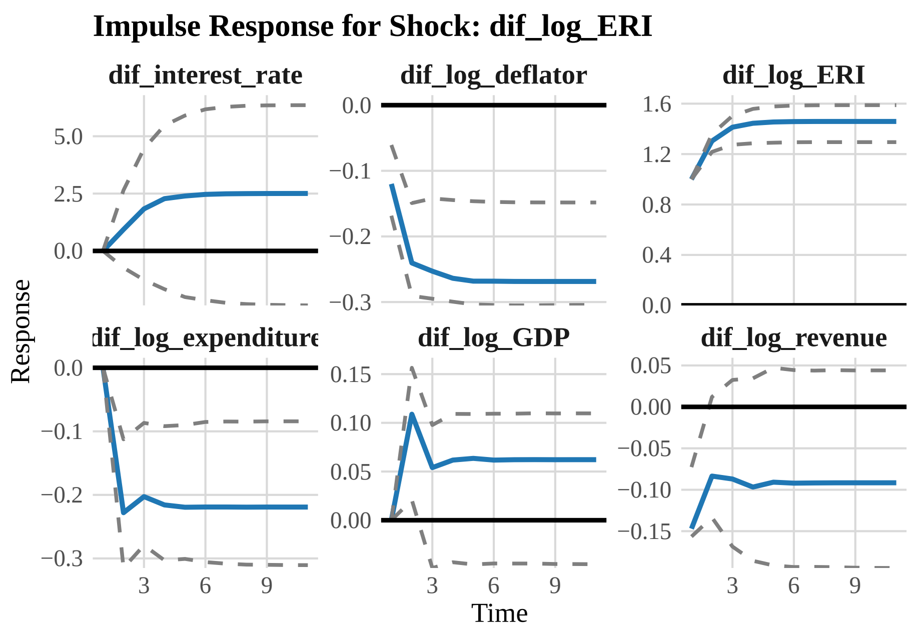


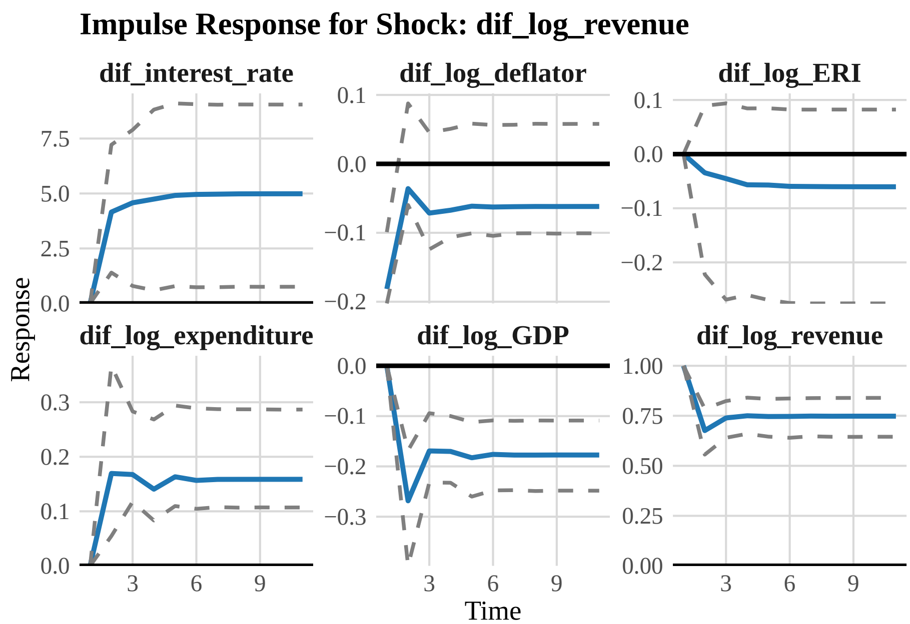
Impulse Response for Shock: dif_log_expenditure
dif_interest_rate
dif_log_deflator
dif_log_ERI
dif_log_expenditure
dif_log_GDP
dif_log_revenue
Response
Time

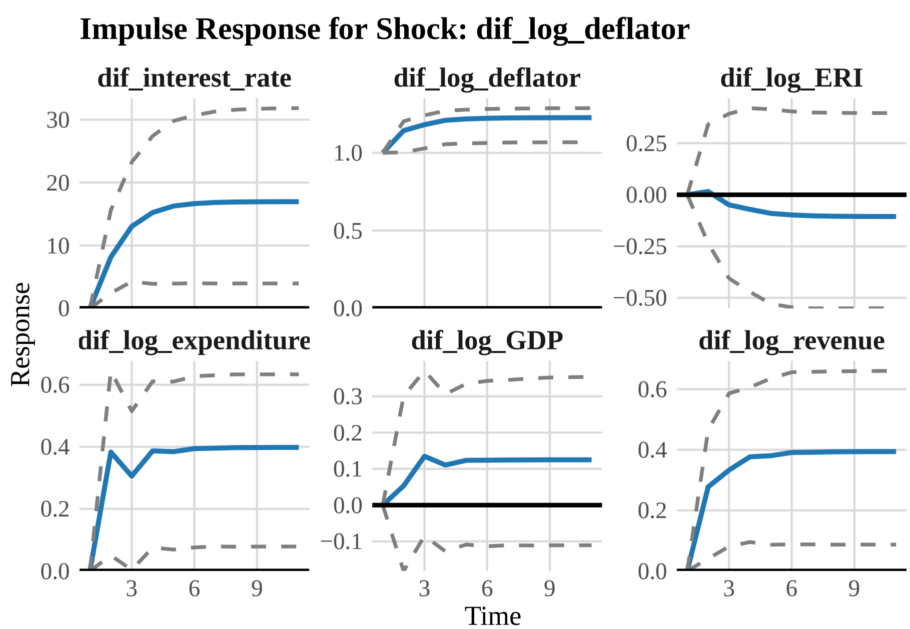


Impulse Response for Shock: dif_interest_rate
dif_interest_rate
dif_log_deflator
dif_log_ERI
dif_log_expenditure
dif_log_GDP
dif_log_revenue
Response
Time

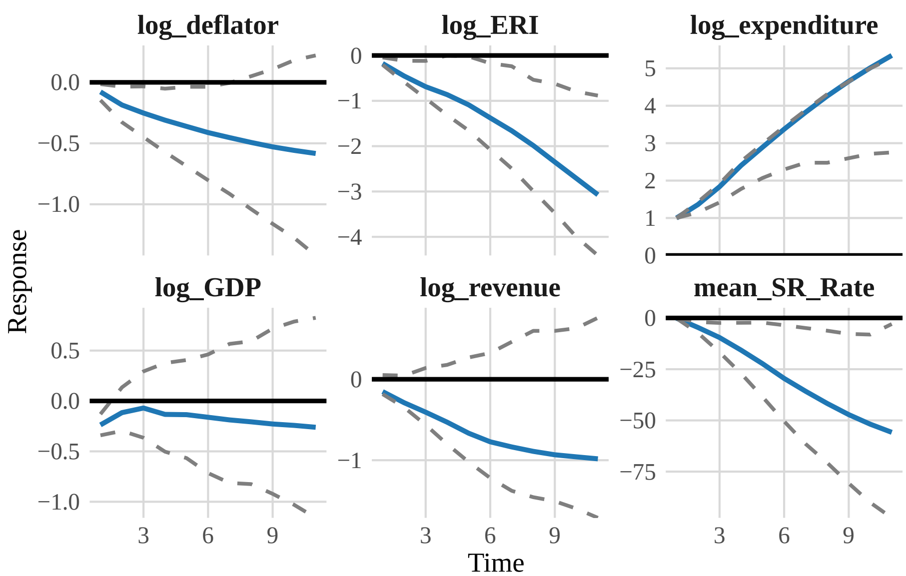

Impulse Response for Shock: dif_log_GDP
dif_interest_rate
dif_log_deflator
dif_log_ERI
dif_log_expenditure
dif_log_GDP
dif_log_revenue
Response
Time


Impulse Response for Shock: dif_log_ERI
dif_interest_rate
dif_log_deflator
dif_log_ERI
dif_log_expenditure
dif_log_GDP
dif_log_revenue
Response
Time

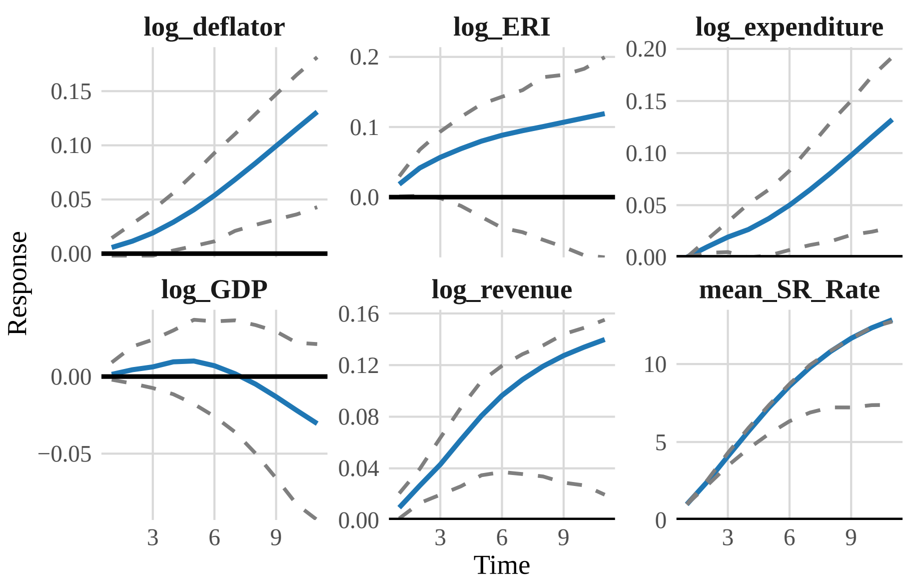

Impulse Response for Shock: dif_log_revenue
dif_interest_rate
dif_log_deflator
dif_log_ERI
dif_log_expenditure
dif_log_GDP
dif_log_revenue
Response
Time

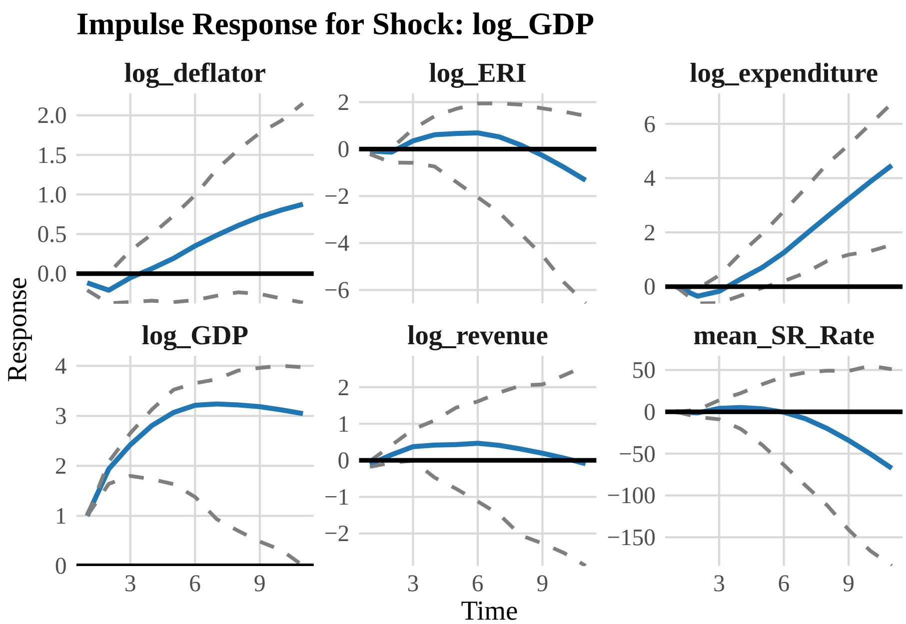

Impulse Response for Shock: dif_log_deflator
dif_interest_rate
dif_log_deflator
dif_log_ERI
dif_log_expenditure
dif_log_GDP
dif_log_revenue
Response
Time

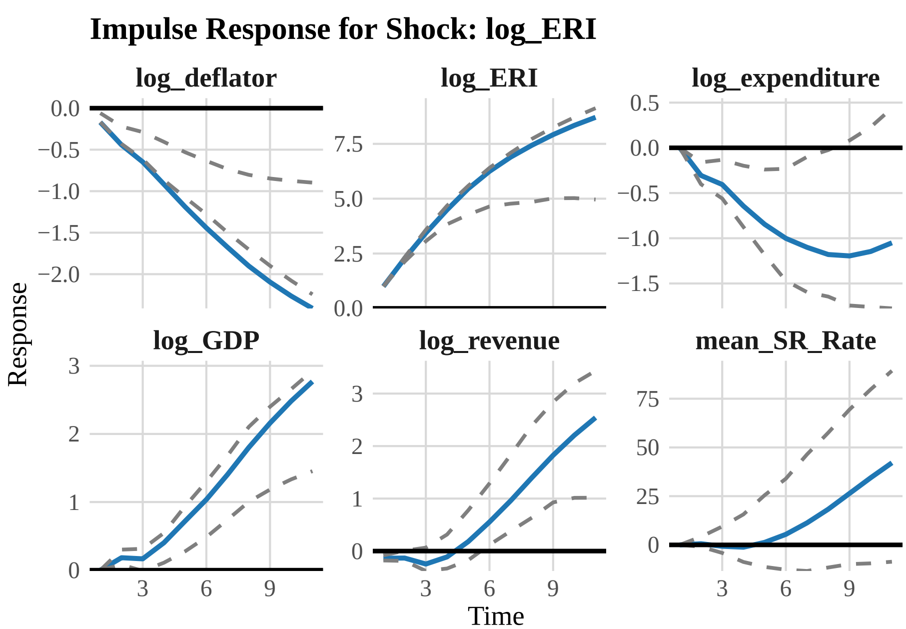
Impulse Response for Shock: log_expenditure

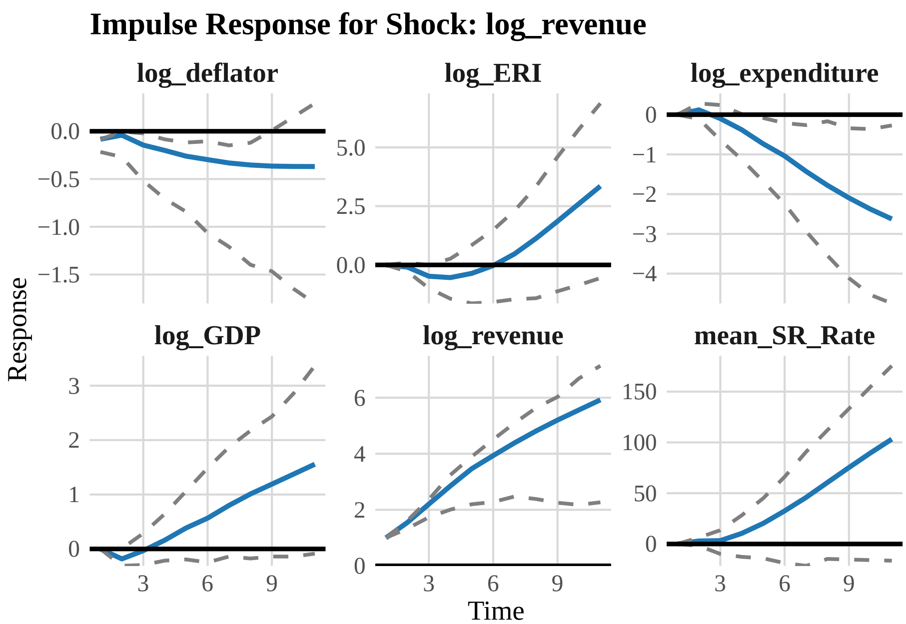


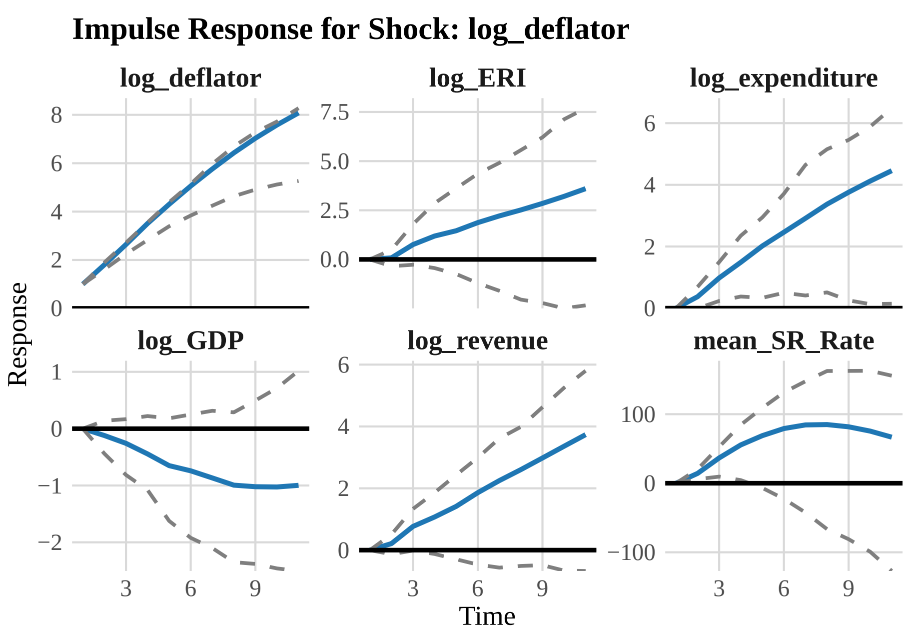
Impulse Response for Shock: mean_SR_Rate




Impulse Response for Shock: log_GDP
log_deflator
log_ERI
log_expenditure
log_GDP
log_revenue
mean_SR_Rate
Response
Time


Impulse Response for Shock: log_ERI
log_deflator
log_ERI
log_expenditure
log_GDP
log_revenue
mean_SR_Rate
Response
Time






```
## $dif_log_expenditure
##       dif_log_expenditure dif_interest_rate dif_log_GDP dif_log_ERI
##  [1,]           1.0000000      0.000000e+00  0.00000000  0.00000000
##  [2,]           0.7685454      7.187336e-06  0.09519169  0.03116201
##  [3,]           0.7467714      9.497049e-06  0.12006161  0.02995596
```

```
##  [4,]           0.7428601      9.884356e-06  0.12060018  0.02985645
##  [5,]           0.7426351      9.963489e-06  0.12056429  0.02985451
##  [6,]           0.7425661      9.998599e-06  0.12056775  0.02985166
##  [7,]           0.7425636      1.000183e-05  0.12056768  0.02985153
##  [8,]           0.7425625      1.000298e-05  0.12056738  0.02985148
##  [9,]           0.7425624      1.000316e-05  0.12056746  0.02985148
## [10,]           0.7425624      1.000319e-05  0.12056745  0.02985148
##       dif_log_revenue dif_log_deflator
##  [1,]      0.00000000       0.00000000
##  [2,]      0.01719688       0.08789681
##  [3,]      0.01633214       0.08686935
##  [4,]      0.01663430       0.09003907
##  [5,]      0.01692168       0.09001443
##  [6,]      0.01694500       0.09005944
##  [7,]      0.01694686       0.09006036
##  [8,]      0.01694682       0.09006180
##  [9,]      0.01694682       0.09006181
## [10,]      0.01694682       0.09006187
##
## $dif_interest_rate
##       dif_log_expenditure dif_interest_rate dif_log_GDP dif_log_ERI
##  [1,]        0.0007707816       0.999229218  0.00000000 0.000000000
##  [2,]        0.0815221975       0.012231529  0.04730402 0.008865359
##  [3,]        0.0758539781       0.009909552  0.04700427 0.013326340
##  [4,]        0.0732458273       0.009569436  0.04562426 0.014310555
##  [5,]        0.0733363531       0.009492191  0.04521856 0.014280539
##  [6,]        0.0732642290       0.009481937  0.04522244 0.014299898
##  [7,]        0.0732555003       0.009479590  0.04521029 0.014300440
##  [8,]        0.0732540537       0.009479231  0.04520852 0.014300442
##  [9,]        0.0732537378       0.009479161  0.04520830 0.014300461
## [10,]        0.0732536734       0.009479149  0.04520824 0.014300462
##       dif_log_revenue dif_log_deflator
##  [1,]       0.0000000        0.0000000
##  [2,]       0.1751375        0.6749394
##  [3,]       0.1383542        0.7155516
##  [4,]       0.1330447        0.7242052
##  [5,]       0.1320440        0.7256283
##  [6,]       0.1318919        0.7258396
##  [7,]       0.1318559        0.7258983
##  [8,]       0.1318515        0.7259062
##  [9,]       0.1318504        0.7259079
## [10,]       0.1318503        0.7259082
##
## $dif_log_GDP
##       dif_log_expenditure dif_interest_rate dif_log_GDP dif_log_ERI
##  [1,]          0.04341083      8.496477e-05   0.9565042 0.000000000
```

```
##  [2,]          0.11501119      7.094994e-05   0.8165056 0.009347573
##  [3,]          0.12288543      6.683527e-05   0.7981181 0.011040115
##  [4,]          0.12286461      6.672582e-05   0.7980011 0.011036674
##  [5,]          0.12283407      6.671742e-05   0.7978092 0.011036117
##  [6,]          0.12283305      6.671386e-05   0.7977799 0.011037749
##  [7,]          0.12283226      6.671453e-05   0.7977796 0.011037787
##  [8,]          0.12283229      6.671455e-05   0.7977795 0.011037789
##  [9,]          0.12283233      6.671456e-05   0.7977794 0.011037792
## [10,]          0.12283234      6.671456e-05   0.7977794 0.011037793
##       dif_log_revenue dif_log_deflator
##  [1,]      0.00000000      0.000000000
##  [2,]      0.05682787      0.002236839
##  [3,]      0.06084046      0.007049088
##  [4,]      0.06057845      0.007452472
##  [5,]      0.06067920      0.007574729
##  [6,]      0.06070821      0.007574378
##  [7,]      0.06070909      0.007574514
##  [8,]      0.06070907      0.007574606
##  [9,]      0.06070909      0.007574634
## [10,]      0.06070909      0.007574634
##
## $dif_log_ERI
##       dif_log_expenditure dif_interest_rate dif_log_GDP dif_log_ERI
##  [1,]          0.01044730      0.0002103077  0.01215989   0.9771825
##  [2,]          0.02764119      0.0001923101  0.01756682   0.9533415
##  [3,]          0.02733142      0.0001899561  0.01735067   0.9501671
##  [4,]          0.02737985      0.0001901782  0.01750149   0.9494592
##  [5,]          0.02737348      0.0001903321  0.01749805   0.9491411
##  [6,]          0.02737173      0.0001903832  0.01749817   0.9490880
##  [7,]          0.02737180      0.0001903966  0.01749817   0.9490716
##  [8,]          0.02737173      0.0001903996  0.01749818   0.9490688
##  [9,]          0.02737174      0.0001904003  0.01749817   0.9490681
## [10,]          0.02737174      0.0001904004  0.01749818   0.9490680
##       dif_log_revenue dif_log_deflator
##  [1,]     0.000000000     0.0000000000
##  [2,]     0.001036902     0.0002213094
##  [3,]     0.001121619     0.0038392439
##  [4,]     0.001232706     0.0042366143
##  [5,]     0.001232342     0.0045647348
##  [6,]     0.001237200     0.0046144817
##  [7,]     0.001237266     0.0046308029
##  [8,]     0.001237352     0.0046334947
##  [9,]     0.001237359     0.0046341995
## [10,]     0.001237362     0.0046343109
##
## $dif_log_revenue
```

```
##       dif_log_expenditure dif_interest_rate dif_log_GDP dif_log_ERI
##  [1,]          0.01365033      0.0001065250 0.005391315  0.02074945
##  [2,]          0.01064955      0.0001260689 0.089035357  0.01911693
##  [3,]          0.01110409      0.0001266540 0.093109489  0.01891940
##  [4,]          0.01252946      0.0001273533 0.092841508  0.01893105
##  [5,]          0.01271779      0.0001274609 0.093057887  0.01894720
##  [6,]          0.01274838      0.0001274693 0.093077917  0.01894537
##  [7,]          0.01274860      0.0001274747 0.093079538  0.01894533
##  [8,]          0.01274865      0.0001274757 0.093079378  0.01894531
##  [9,]          0.01274865      0.0001274758 0.093079374  0.01894530
## [10,]          0.01274866      0.0001274759 0.093079371  0.01894530
##       dif_log_revenue dif_log_deflator
##  [1,]       0.9601024       0.00000000
##  [2,]       0.8236621       0.05741003
##  [3,]       0.8176931       0.05904722
##  [4,]       0.8152851       0.06028558
##  [5,]       0.8148852       0.06026442
##  [6,]       0.8147614       0.06033942
##  [7,]       0.8147595       0.06033953
##  [8,]       0.8147580       0.06034120
##  [9,]       0.8147579       0.06034128
## [10,]       0.8147579       0.06034132
##
## $dif_log_deflator
##       dif_log_expenditure dif_interest_rate dif_log_GDP dif_log_ERI
##  [1,]        0.0005115897      4.447211e-06  0.02308135  0.01342440
##  [2,]        0.0086805679      6.152800e-06  0.02449600  0.02531803
##  [3,]        0.0095742744      7.745180e-06  0.03039659  0.02521146
##  [4,]        0.0095727031      7.959745e-06  0.03058106  0.02528640
##  [5,]        0.0095897598      8.046781e-06  0.03057729  0.02529972
##  [6,]        0.0095909850      8.069283e-06  0.03058662  0.02529904
##  [7,]        0.0095913239      8.073152e-06  0.03058699  0.02529899
##  [8,]        0.0095913504      8.074111e-06  0.03058701  0.02529898
##  [9,]        0.0095913489      8.074304e-06  0.03058702  0.02529898
## [10,]        0.0095913505      8.074338e-06  0.03058702  0.02529898
##       dif_log_revenue dif_log_deflator
##  [1,]      0.03072389        0.9322543
##  [2,]      0.04745584        0.8940434
##  [3,]      0.04809201        0.8867179
##  [4,]      0.04805722        0.8864947
##  [5,]      0.04808086        0.8864443
##  [6,]      0.04808088        0.8864344
##  [7,]      0.04808079        0.8864338
##  [8,]      0.04808081        0.8864338
##  [9,]      0.04808081        0.8864338
## [10,]      0.04808081        0.8864338
```

```
## $JB
##
##  JB-Test (multivariate)
##
## data:  Residuals of VAR object reduced_VAR
## Chi-squared = 15539, df = 12, p-value < 2.2e-16
##
##
## $Skewness
##
##  Skewness only (multivariate)
##
## data:  Residuals of VAR object reduced_VAR
## Chi-squared = 763.74, df = 6, p-value < 2.2e-16
##
##
## $Kurtosis
##
##  Kurtosis only (multivariate)
##
## data:  Residuals of VAR object reduced_VAR
## Chi-squared = 14775, df = 6, p-value < 2.2e-16
```

6.3 Chow test for Structural Breaks

Interested in assessing differences in the fiscal multipliers pre / post GFC

```
## [1] "Year"                "CDID"                "dif_log_expenditure"
## [4] "dif_interest_rate"   "dif_log_GDP"         "dif_log_ERI"
## [7] "dif_log_revenue"     "dif_log_deflator"
```

```
## [1] 0.01346513
```

```
## [1] -0.03959656
```

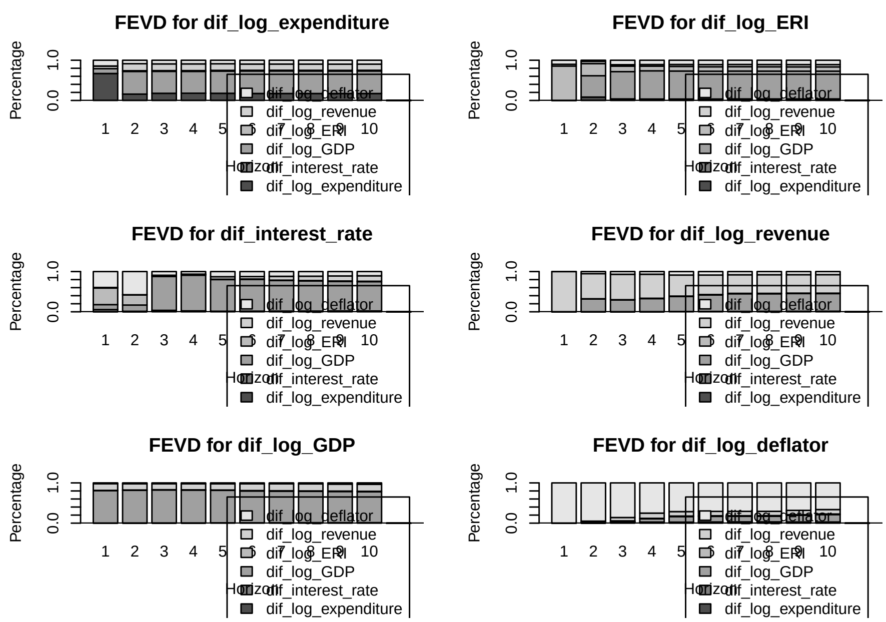
```
## [1] NA
```

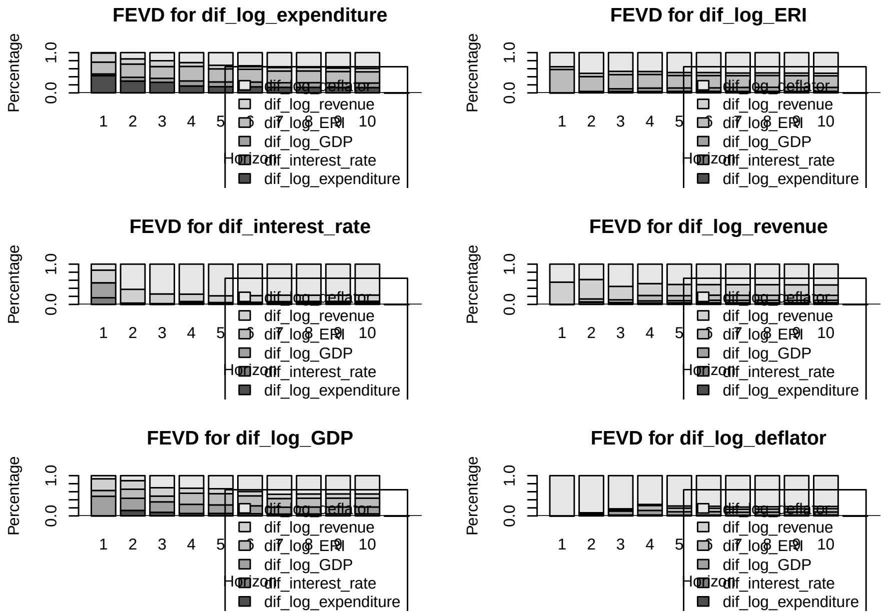
```
## [1] 0.004722417
```

```
## [1] 0.01282792
```

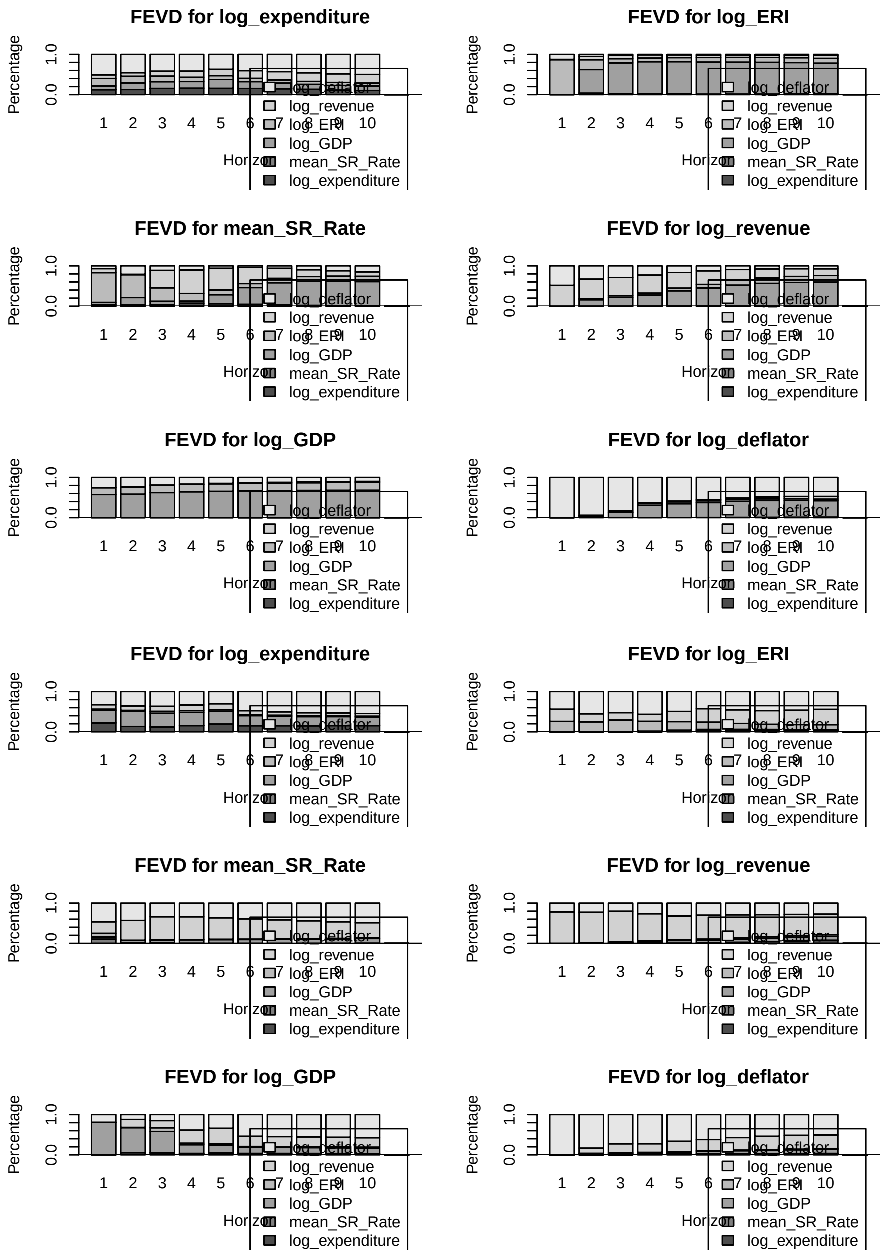
```
## [1] NA
```

```
## $OptimalLag_Pre2008
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      1      1      1      1
##
## $OptimalLag_Post2008
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      5      1      1      5
##
##  [1] 0.8459462 0.8459462 0.8290196 0.8213950 0.8213950 0.7956707 0.7898478
##  [8] 0.7898478 0.7634168 0.7634168 0.7522628 0.7522628 0.7426912 0.7426912
## [15] 0.7351169 0.7351169 0.6660874 0.6660874 0.6624267 0.6624267 0.6333053
## [22] 0.4105192 0.4105192 0.1652262
##  [1] 0.8685553 0.8482476 0.8482476 0.8294580 0.8294580 0.8280265 0.8280265
##  [8] 0.7972945 0.7972945 0.7834309 0.7834309 0.7476465 0.7476465 0.7444801
## [15] 0.7444801 0.7399308 0.7399308 0.5326667 0.5326667 0.4076156 0.4076156
## [22] 0.4039885 0.2519929 0.2519929
```

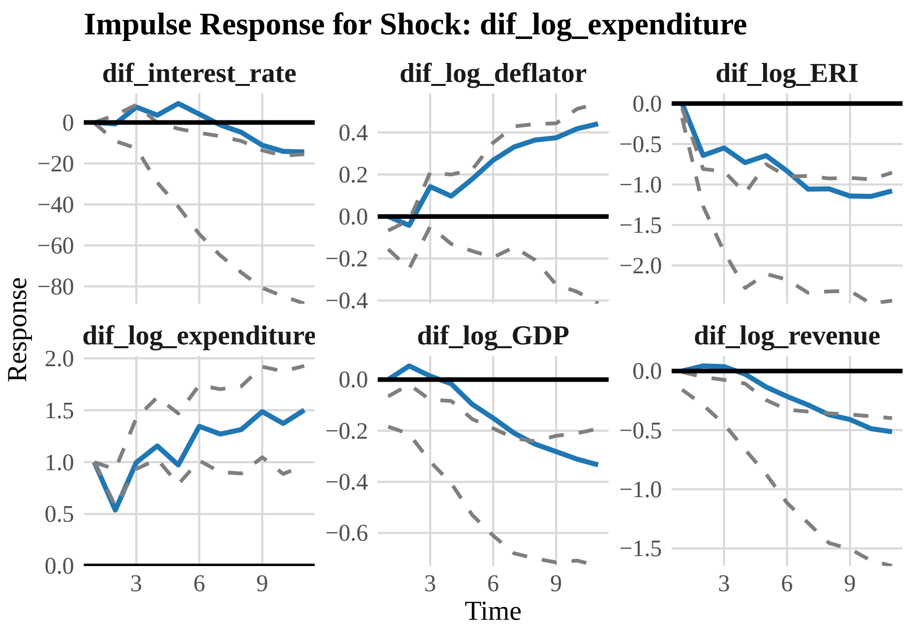


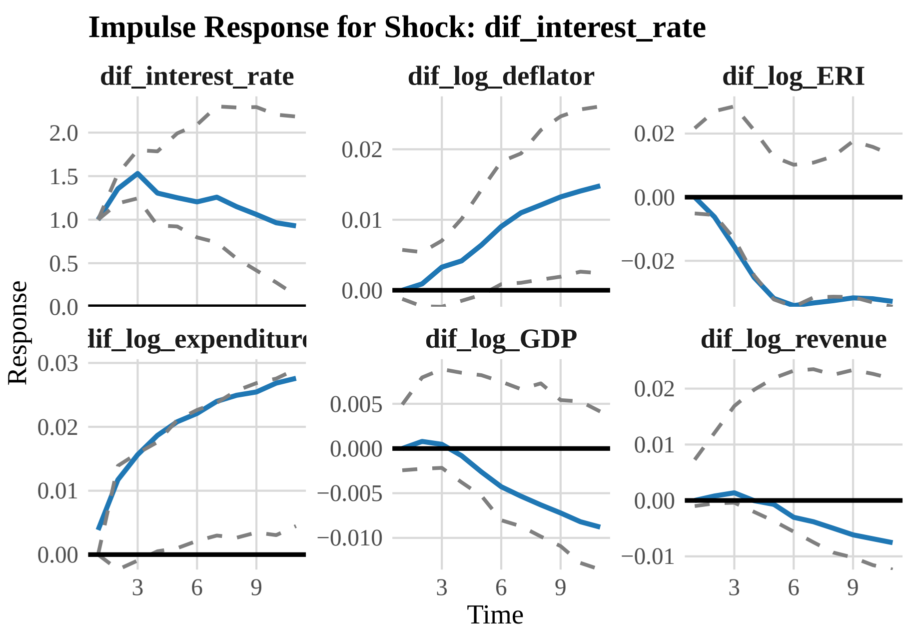


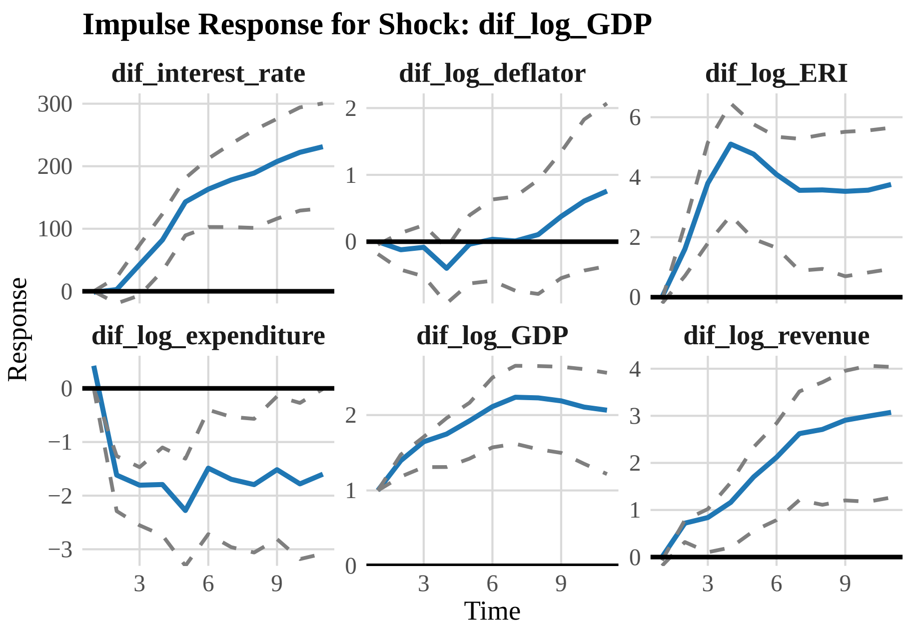

```
## $OptimalLag_Pre2008
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      4      2      1      4
##
## $OptimalLag_Post2008
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      5      2      1      5
##
##  [1] 0.99534575 0.95363707 0.95363707 0.94139795 0.94139795 0.90398895
##  [7] 0.90398895 0.82808169 0.82808169 0.79998735 0.79998735 0.75422100
## [13] 0.75422100 0.74924185 0.74924185 0.69753139 0.67194628 0.67194628
## [19] 0.54721720 0.54569288 0.54569288 0.39042514 0.39042514 0.03022064
##  [1] 1.0204611 0.9388997 0.9388997 0.9174260 0.9174260 0.8273002 0.8273002
##  [8] 0.7800541 0.7650301 0.7409686 0.7409686 0.7042138 0.7042138 0.6956662
## [15] 0.6956662 0.6838859 0.6838859 0.6400863 0.6400863 0.6210795 0.6210795
## [22] 0.5437732 0.2156430 0.2156430
```

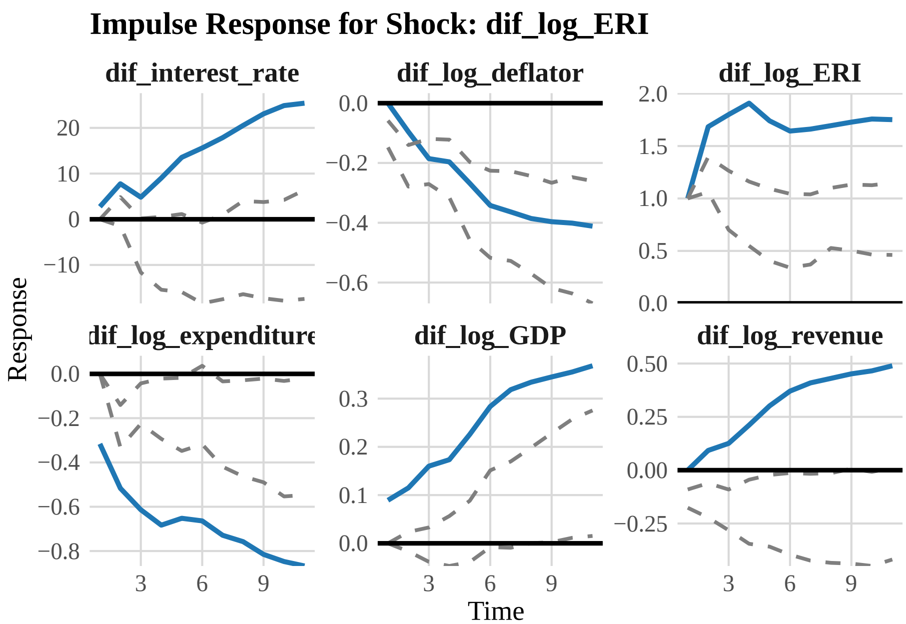

FEVD for log_expenditure
FEVD for log_ERI
FEVD for mean_SR_Rate
FEVD for log_revenue
FEVD for log_GDP
FEVD for log_deflator
FEVD for log_expenditure
FEVD for log_ERI
FEVD for mean_SR_Rate
FEVD for log_revenue
FEVD for log_GDP
FEVD for log_deflator
Percentage
0.0
1.0
1 2 3 4 5 6 7 8 9 10
Horizon
log_deflator
log_revenue
log_ERI
log_GDP
mean_SR_Rate
log_expenditure

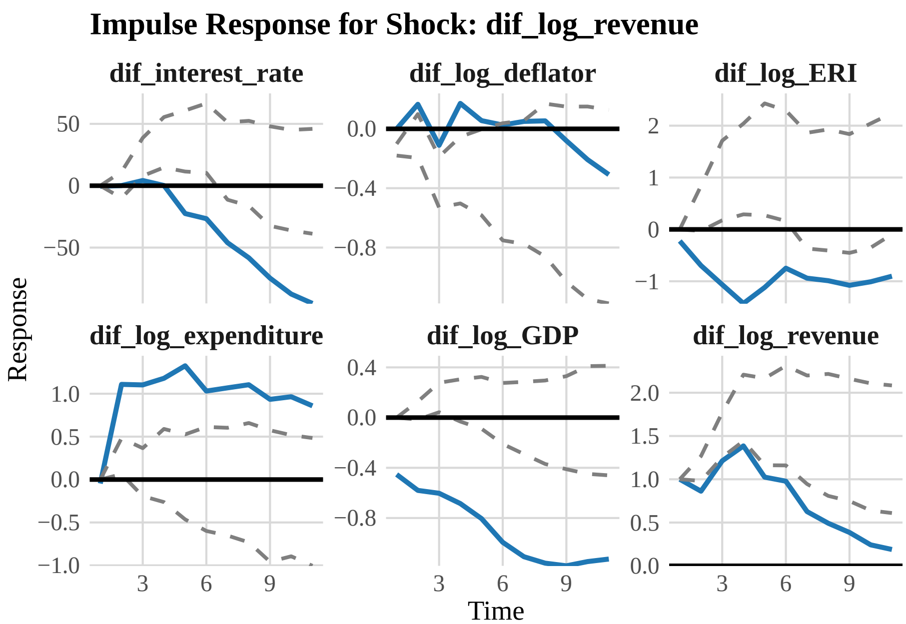
6.3.1 Pre GFC Shocks

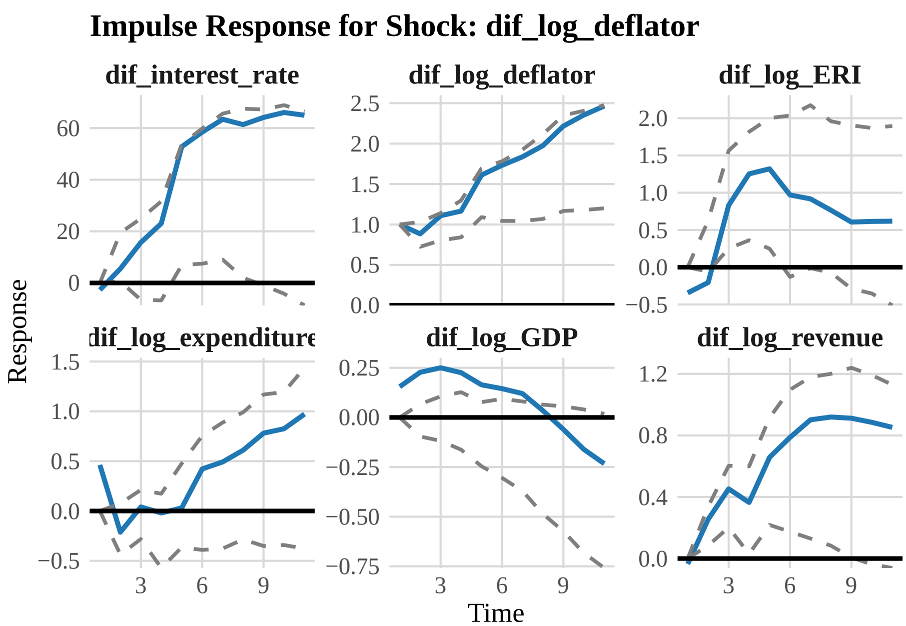


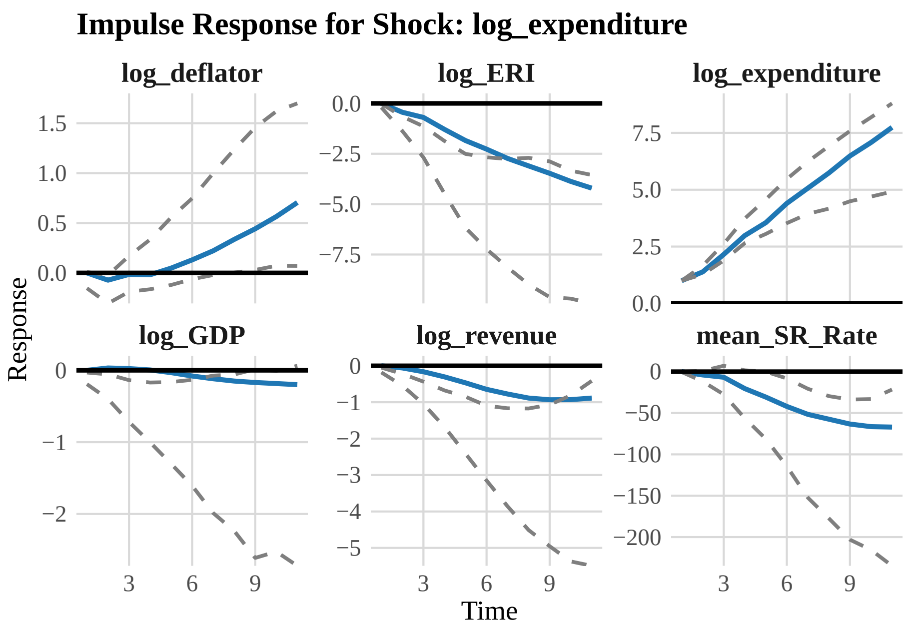


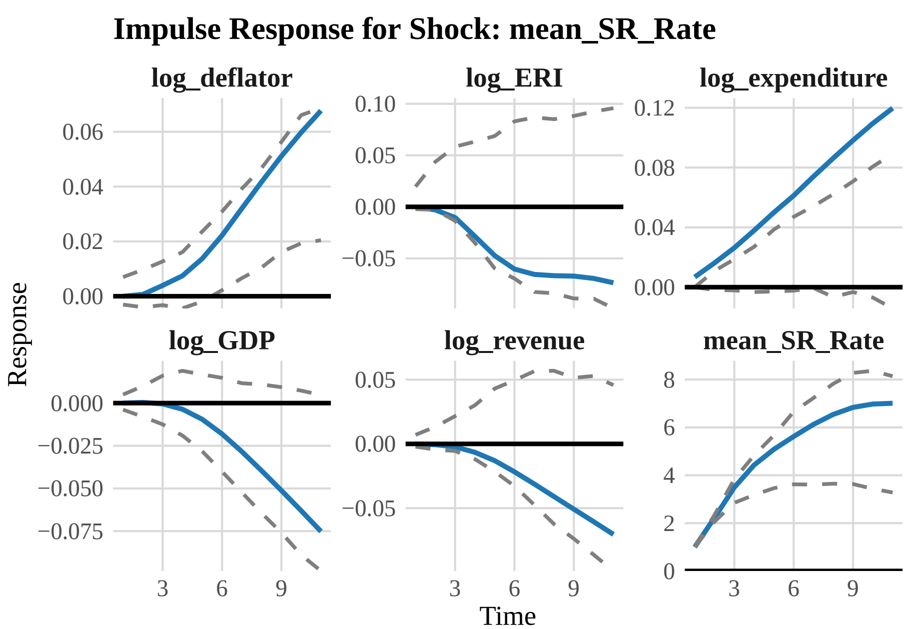

Impulse Response for Shock: dif_log_GDP
dif_interest_rate
dif_log_deflator
dif_log_ERI
dif_log_expenditure
dif_log_GDP
dif_log_revenue
Response
Time


Impulse Response for Shock: dif_log_ERI
dif_interest_rate
dif_log_deflator
dif_log_ERI
dif_log_expenditure
dif_log_GDP
dif_log_revenue
Response
Time

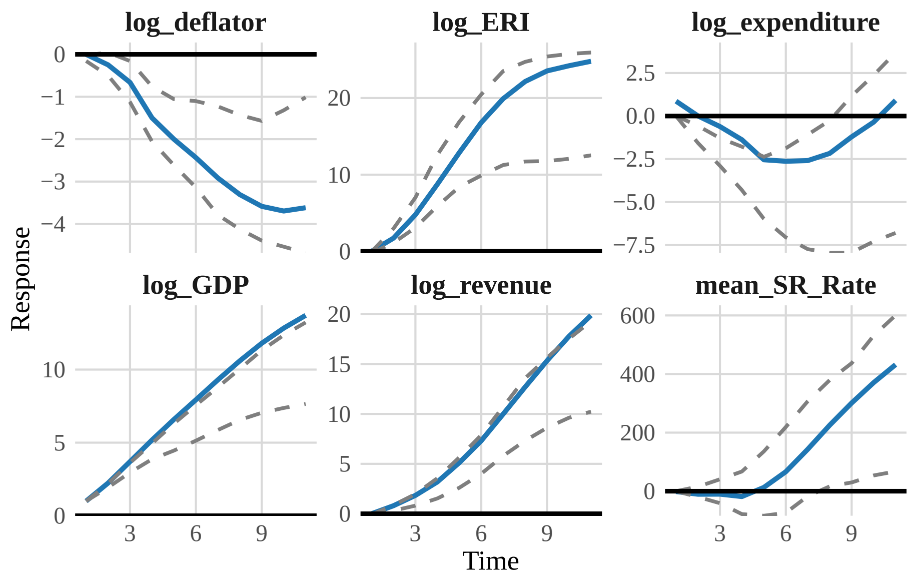

Impulse Response for Shock: dif_log_revenue
dif_interest_rate
dif_log_deflator
dif_log_ERI
dif_log_expenditure
dif_log_GDP
dif_log_revenue
Response
Time


Impulse Response for Shock: dif_log_deflator
dif_interest_rate
dif_log_deflator
dif_log_ERI
dif_log_expenditure
dif_log_GDP
dif_log_revenue
Response
Time

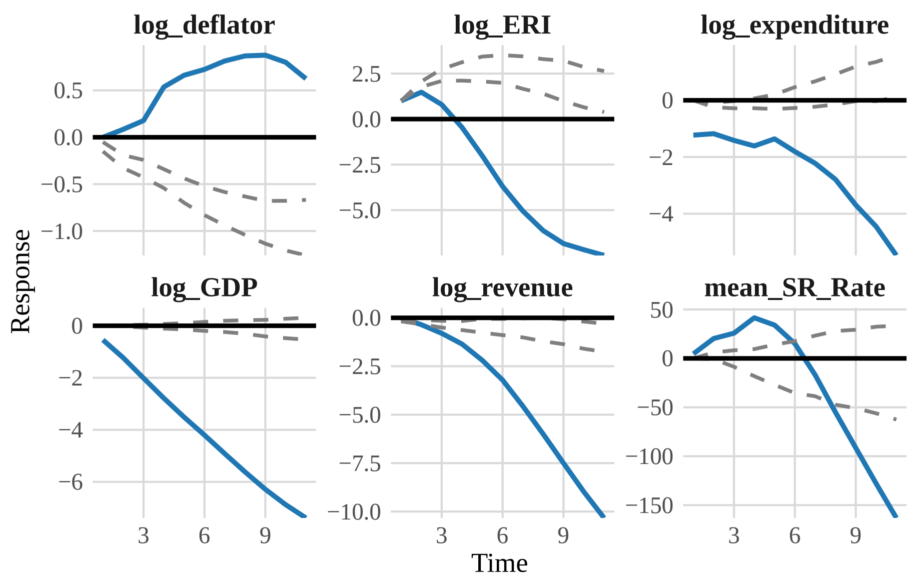

Impulse Response for Shock: log_expenditure
log_deflator
log_ERI
log_expenditure
log_GDP
log_revenue
mean_SR_Rate
Response
Time

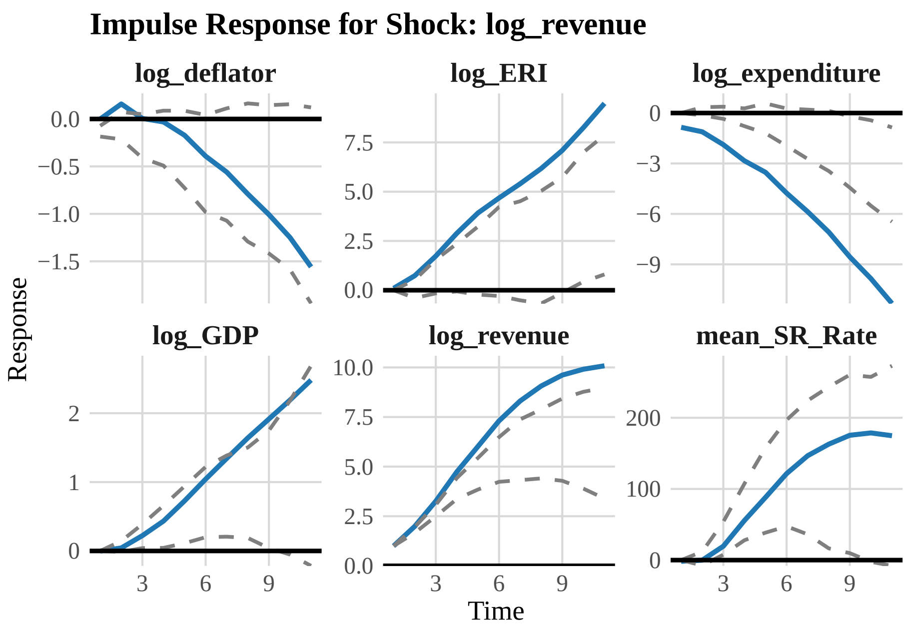

Impulse Response for Shock: mean_SR_Rate
log_deflator
log_ERI
log_expenditure
log_GDP
log_revenue
mean_SR_Rate
Response
Time

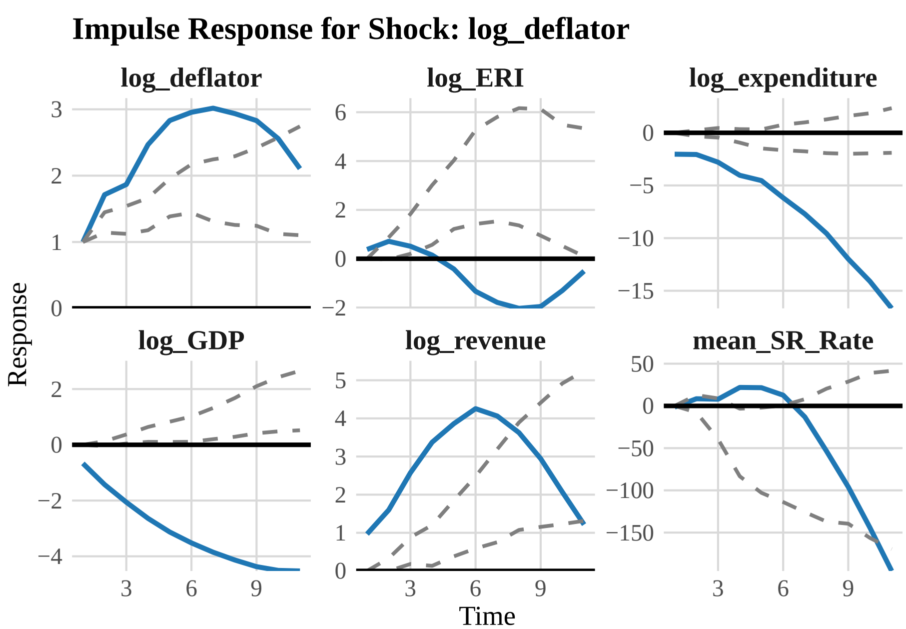
Impulse Response for Shock: log_GDP



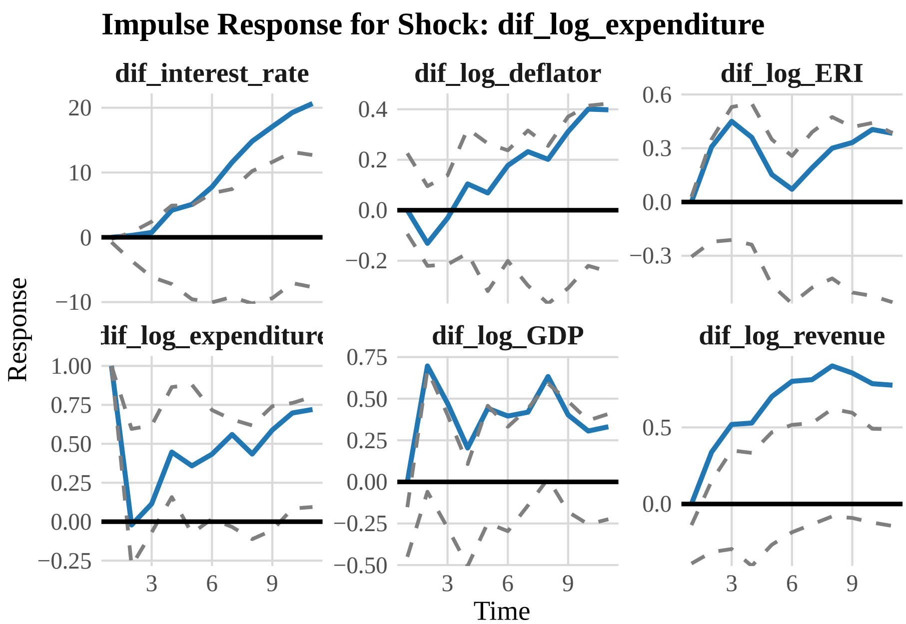
Impulse Response for Shock: log_ERI

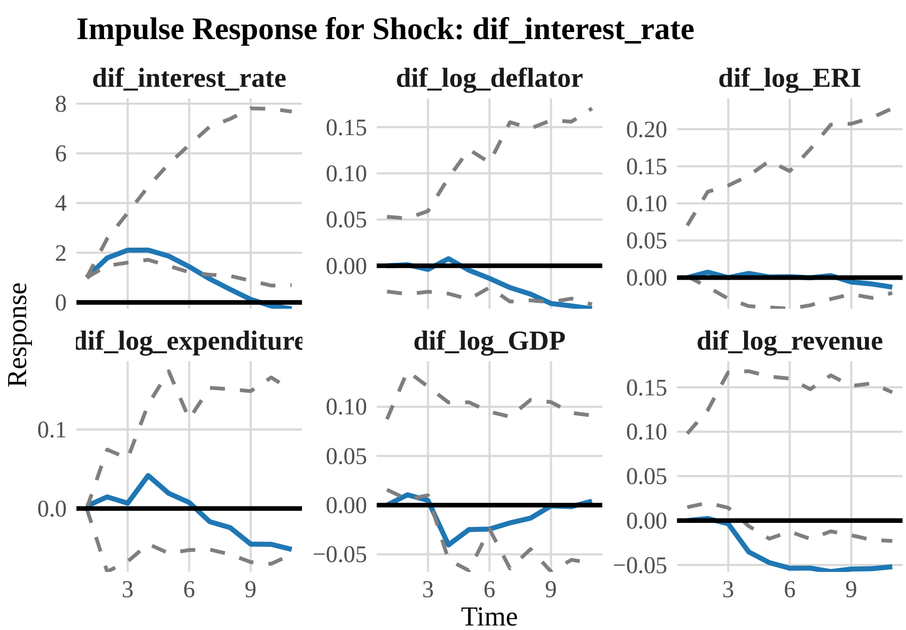


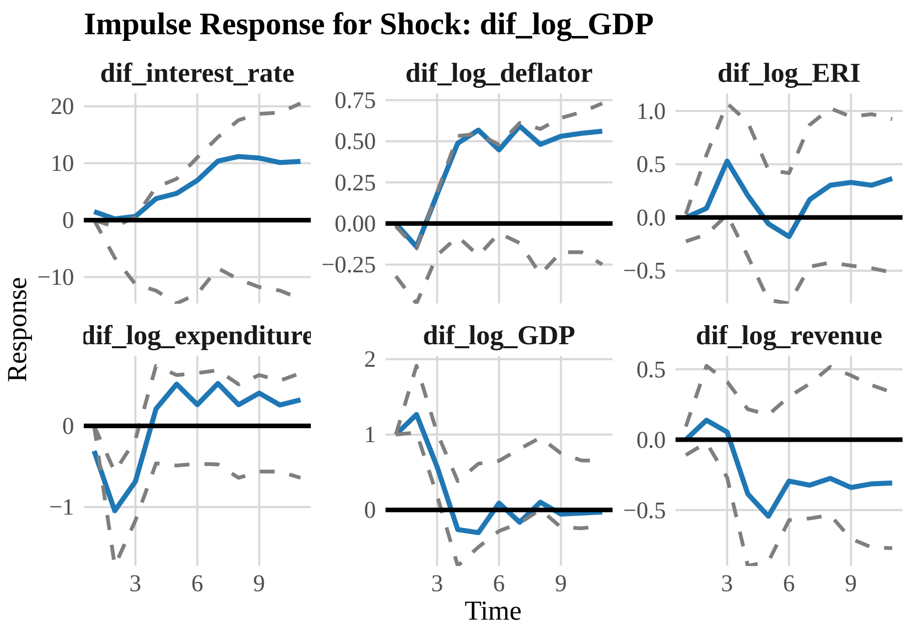

Impulse Response for Shock: log_revenue
log_deflator
log_ERI
log_expenditure
log_GDP
log_revenue
mean_SR_Rate
Response
Time

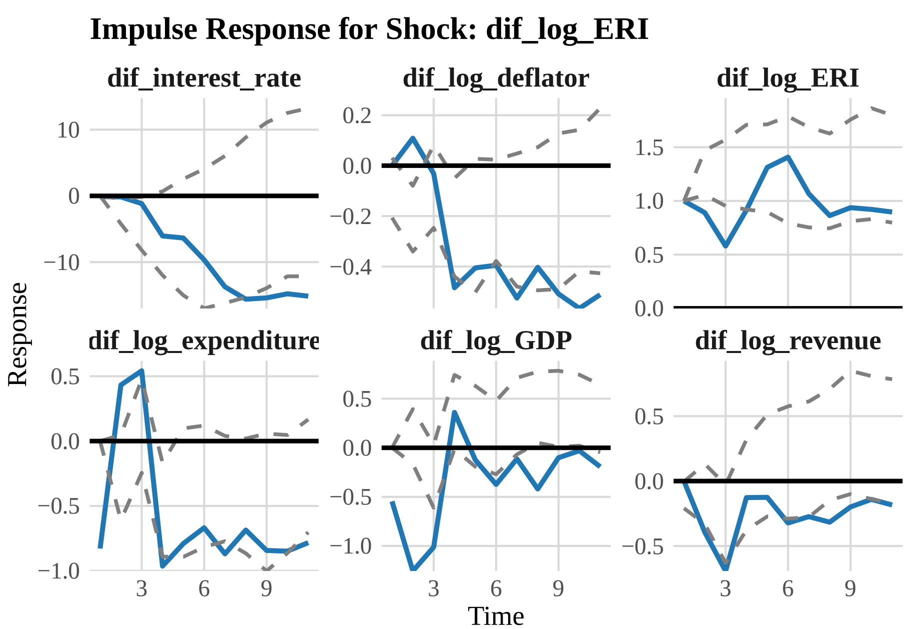

Impulse Response for Shock: log_deflator
log_deflator
log_ERI
log_expenditure
log_GDP
log_revenue
mean_SR_Rate
Response
Time

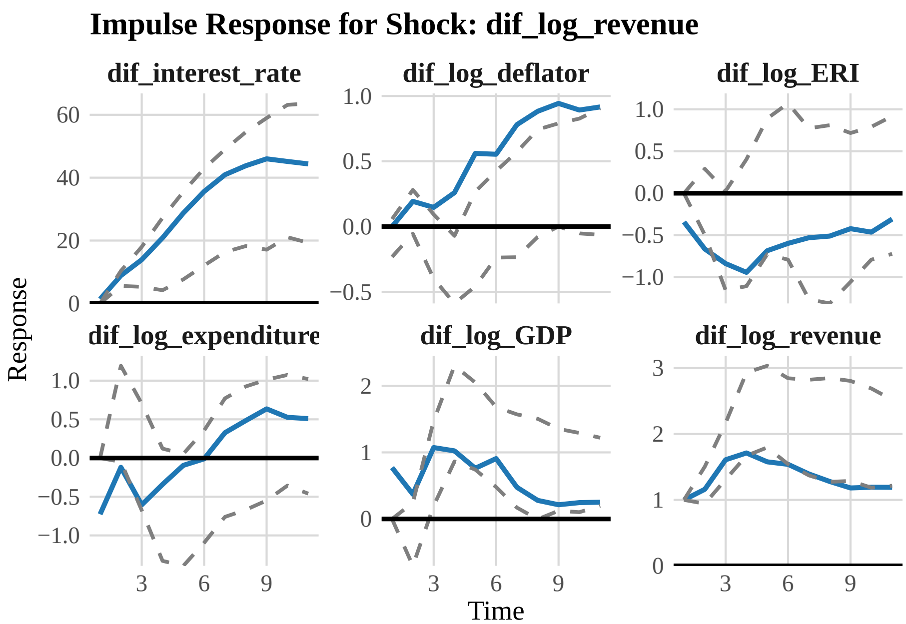
6.3.2 Post GFC Shocks

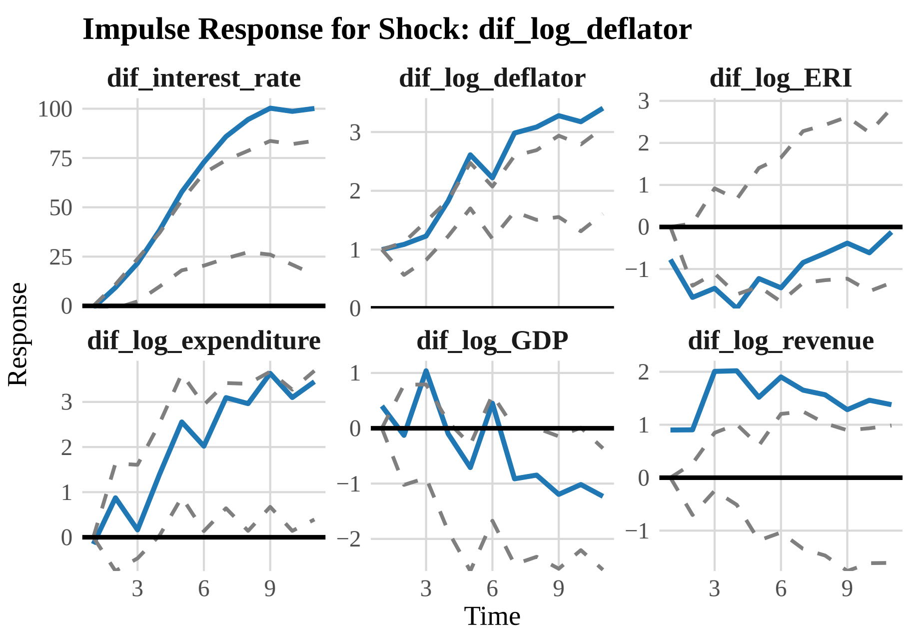




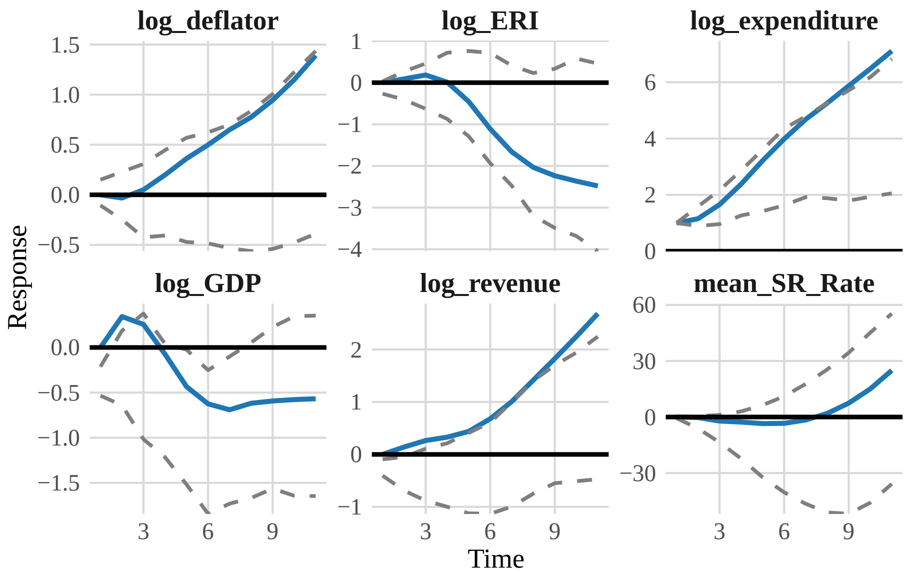

Impulse Response for Shock: dif_log_GDP
dif_interest_rate
dif_log_deflator
dif_log_ERI
dif_log_expenditure
dif_log_GDP
dif_log_revenue
Response
Time
20
10
0
−10
0.75
0.50
0.25
0.00
−0.25
1.0
0.5
0.0
−0.5
−1
2
1
3
6
9


Impulse Response for Shock: dif_log_ERI
dif_interest_rate
dif_log_deflator
dif_log_ERI
dif_log_expenditure
dif_log_GDP
dif_log_revenue
Response
Time
10
0
−10
0.2
0.0
−0.2
−0.4
1.5
1.0
0.5
−0.5
−1.0
3
6
9

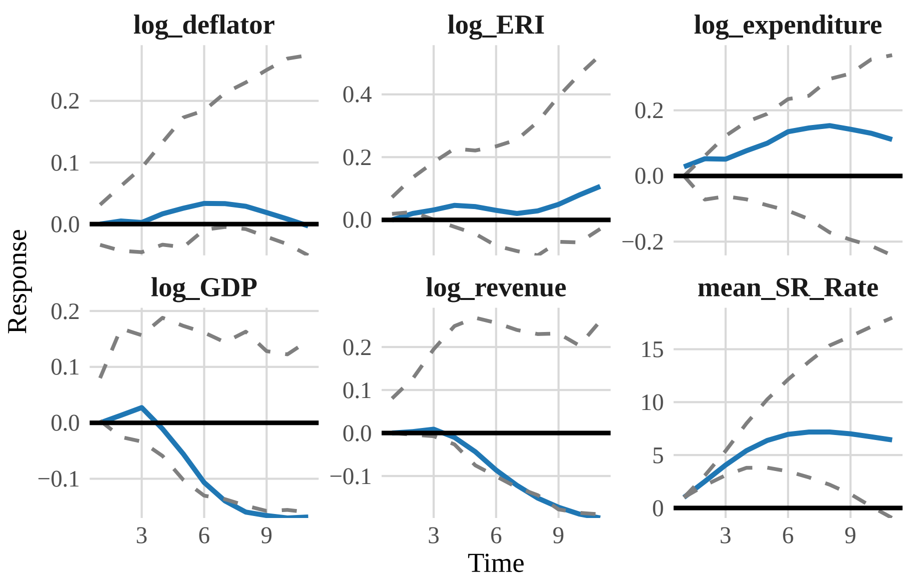

Impulse Response for Shock: dif_log_revenue
dif_interest_rate
dif_log_deflator
dif_log_ERI
dif_log_expenditure
dif_log_GDP
dif_log_revenue
Response
Time


Impulse Response for Shock: dif_log_deflator
dif_interest_rate
dif_log_deflator
dif_log_ERI
dif_log_expenditure
dif_log_GDP
dif_log_revenue
Response
Time

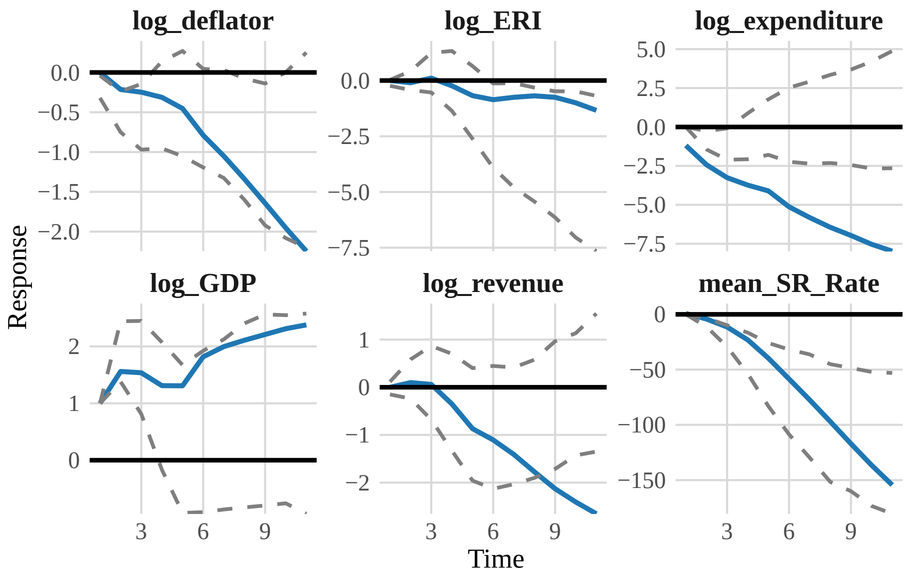
Impulse Response for Shock: log_expenditure



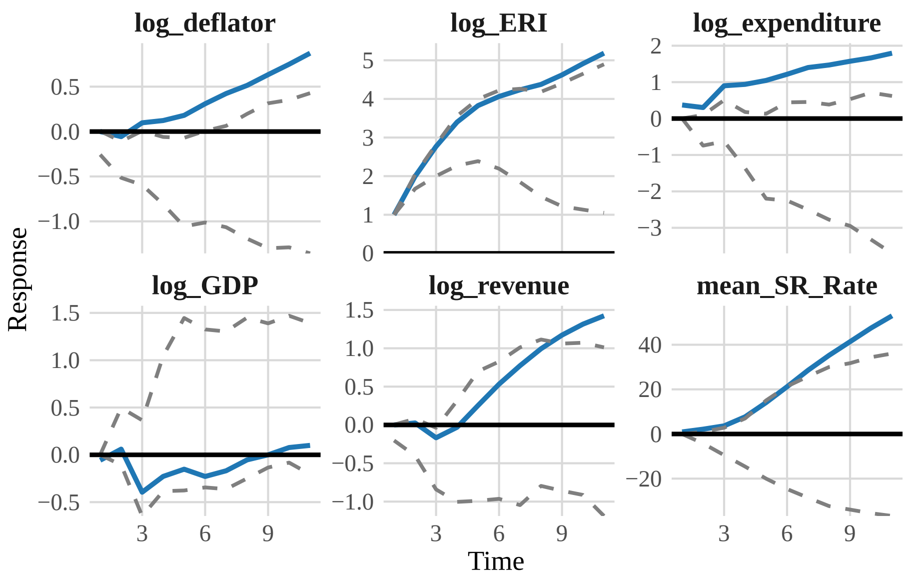
Impulse Response for Shock: mean_SR_Rate

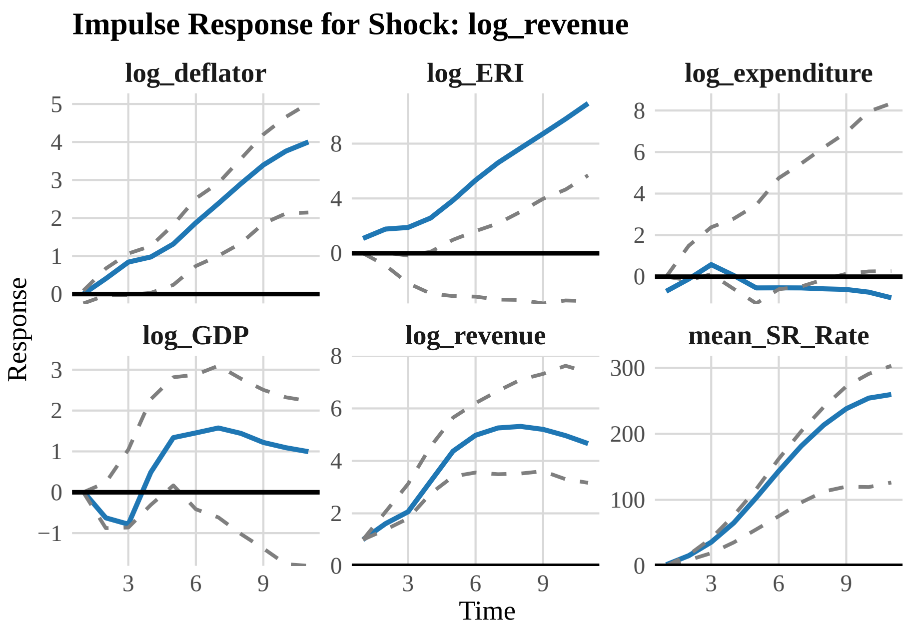


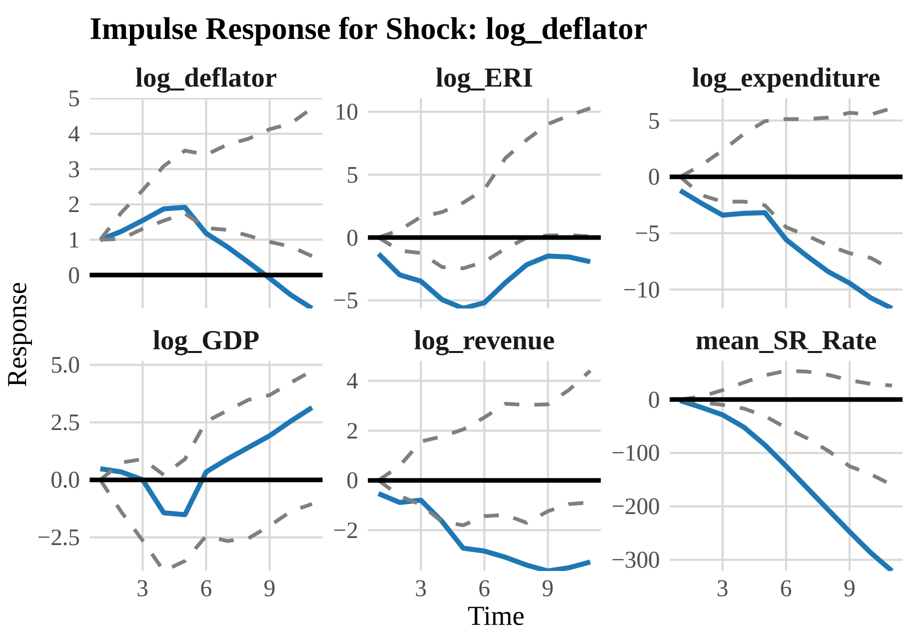
Impulse Response for Shock: log_GDP



Impulse Response for Shock: log_ERI







Ramey (2019) define fiscal multipliers: "change in output due to a change in spending or taxes", and highlight the risk of ignoring fiscal foresight. Gechert (2017)

7 Robustness

7.1 Stability

7.2 Normality

7.3 Lag length

7.4 Identification

8 Discussion/ Policy Implications

9 Conclusion

10 Bibliography

References

[1] O.J. Blanchard and D. Leigh. "Growth forecast errors and fiscal multipliers". In: *American Economic Review* 103.3 (2013), pp. 117–120.

[2] E. Ilzetzki, E.G. Mendoza, and C.A. Végh. "How big (small?) are fiscal multipliers?" In: *Journal of Monetary Economics* 60.2 (2013), pp. 239–254.

[3] D. Sutherland, P. Hoeller, and R. Merola. "Fiscal consolidation: How much, how fast and by what means?" In: (2012).

[4] T. Warmedinger, C.D. Checherita-Westphal, and P.H. De Cos. "Fiscal multipliers and beyond". In: *ECB Occasional Paper* 162 (2015).

11 Technical Appendix

```r
# library(knitr)
# library(stargazer)
# library(clipr)
#library(kableExtra)

library(ggplot2)
library(knitr)
library(ivreg)
library(ggdag)
```

```r
library(data.table)
library(dplyr)
library(tidyr)
library(stargazer)
library(clipr)
library(tibble)
library(lubridate)
# install.packages("seasonal")
library(seasonal)
```

```r
lapply(c("ggplot2", "dplyr", "data.table", "lubridate", "janitor", "broom", "tibble", "tidyr
       require, character.only = TRUE)

# knitr::opts_chunk$set(echo = FALSE)
knitr::opts_chunk$set(echo = FALSE, warning = FALSE, message = FALSE)
```

```r
df <- fread("D:/Samid work/University/KCL - Econ and Policy/Dissertation/Data/GDP.csv", skip
```

```r
# Filter the data frame to exclude rows where the column 'title' matches any of the specifie
```

```r
filtered_df <- df %>%
  # Keep only the quarterly data
  filter(nchar(CDID) == 7 & substr(CDID, 6, 6) == "Q") %>%
  # Select relevant columns and rename them
  dplyr::select(CDID, Deflator = L8GG, GDP = ABMI) %>%
  # Create new columns and convert types
  mutate(
    Year = as.numeric(substr(CDID, 1, 4)),
    Quarter = substr(CDID, 6, 7),
    Q = as.numeric(substr(CDID, 7, 7)),
    Deflator = as.numeric(Deflator),
    GDP = as.numeric(GDP)
  ) %>%
  # Filter by year (can modify for testing)
  filter(Year >= 1987)
```

```
fiscal_raw <- fread("D:/Samid work/University/KCL - Econ and Policy/Dissertation/Data/Fiscal

fiscal_proc <- fiscal_raw %>%
  dplyr::select(Date_ID = Transaction, Revenue = OTR, Expenditure = OTE) %>%
  subset(Date_ID !=  "Dataset identifier code" & Date_ID != "Identifier") %>%
  mutate(Year = as.numeric(gsub("\\D", "", Date_ID)),
         Period = gsub("\\d{4}", "", Date_ID))  %>%
  mutate(
    Q = case_when(
      Period == "Jan to Mar " ~ 1,
      Period == "Apr to Jun " ~ 2,
      Period == "Jul to Sep " ~ 3,
      Period == "Oct to Dec " ~ 4
      ),
    Unique_Period = Year +(Q/4)

  ) %>%
  # Convert to numeric and multiply by 1 million so values as these will later be made into
  mutate(Revenue = as.numeric(gsub(",", "", Revenue) ),
         Expenditure = as.numeric(gsub(",", "", Expenditure )))

# join GDP deflator and GDP data

population <- fread("D:/Samid work/University/KCL - Econ and Policy/Dissertation/Data/Popula
                    skip = 4,
                    header = TRUE) %>%
  subset(`Country Name` == "United Kingdom") %>%
  t() %>%
  as.data.frame() %>%
  rownames_to_column(var = "Year") %>%
  rename(Population = V1 ) %>%
  filter(grepl("^\\d{4}$", Year)) %>%
  mutate(Year = as.numeric(Year),
         Population = as.numeric(Population))

Interest_SR <- fread("D:/Samid work/University/KCL - Econ and Policy/Dissertation/Data/3 mon
  mutate(Date = dmy(Date),
         month = month(Date),
         Year = year(Date),
         Q = case_when(
           month %in% 1:3 ~ 1,
           month %in% 4:6 ~ 2,
           month %in% 7:9 ~ 3,
           month %in% 10:12 ~ 4
         )) %>%
```

```r
  group_by(Year, Q) %>%
  summarize(mean_SR_Rate = mean(Rate, na.rm = TRUE))



SONIA <- fread("D:/Samid work/University/KCL - Econ and Policy/Dissertation/Data/Bank of Eng
  mutate(Date = dmy(Date),
         month = month(Date),
         Year = year(Date),
         Q = case_when(
           month %in% 1:3 ~ 1,
           month %in% 4:6 ~ 2,
           month %in% 7:9 ~ 3,
           month %in% 10:12 ~ 4
         )) %>%
  group_by(Year, Q) %>%
  summarize(mean_SONIA = mean(SONIA, na.rm = TRUE))

Policy_Rate <- fread("D:/Samid work/University/KCL - Econ and Policy/Dissertation/Data/Polic
  mutate(Date = parse_date_time(`Date Changed`, orders = "dmy"),
         Q = quarter(Date),
         Year = year(Date)) %>%
  group_by(Year, Q) %>%
  summarise(mean_SR_Rate = mean(Rate, na.rm = TRUE), .groups = "drop") %>%
  complete(Year = full_seq(Year, 1), Q = 1:4) %>%  # Ensure all Year-Quarter combinations
  arrange(Year, Q) %>%
  fill(mean_SR_Rate, .direction = "down")  # Fill missing rates by propagating the previous

Exports <- fread("D:/Samid work/University/KCL - Econ and Policy/Dissertation/Data/Exports.c
  mutate(Date = dmy(Month),
         month = month(Month),
         Year = year(dmy(Month)),
         Q = case_when(
           month %in% 1:3 ~ 1,
           month %in% 4:6 ~ 2,
           month %in% 7:9 ~ 3,
           month %in% 10:12 ~ 4
         )) %>%
  group_by(Year, Q) %>%
  summarize(Exports = sum(Exports, na.rm = TRUE))

ERI <- fread("D:/Samid work/University/KCL - Econ and Policy/Dissertation/Data/Boe ERI.csv")
  mutate(
         month = month(Date),
         Year = year(dmy(Date)),
```

```
        Q = case_when(
          month %in% 1:3 ~ 1,
          month %in% 4:6 ~ 2,
          month %in% 7:9 ~ 3,
          month %in% 10:12 ~ 4
        )) %>%
  rename(ERI = Value)


data <- fiscal_proc %>%
  left_join(filtered_df, by = c("Q" = "Q", "Year" = "Year")) %>%
  left_join(Interest_SR, by = c("Q" = "Q", "Year" = "Year")) %>%
  left_join(Exports, by = c("Q" = "Q", "Year" = "Year")) %>%
  left_join(population, by = c("Year" = "Year")) %>%
  left_join(ERI, by = c("Q" = "Q", "Year" = "Year")) %>%
# Convert variables to per capita, note revenue, expenditure, and GDP are in £ million so no
  mutate(RevenuePerCapita = (Revenue *10^6) /Population,
         ExpenditurePerCapita = (Expenditure *10^6) /Population,
         GDPPerCapita = (GDP *10^6) /Population) %>%

  # Seasonal Adjustment of data using X-13ARIMA-SEATS
   mutate(Revenue_SA = final(seas(ts(Revenue, start = min(Year), frequency = 4))),
         Expenditure_SA = final(seas(ts(Expenditure, start = min(Year), frequency = 4))),
         GDP_SA = final(seas(ts(GDP, start = min(Year), frequency = 4)))) %>%
 # convert back to numeric
     mutate(
    Revenue_SA = as.numeric(Revenue_SA),
    Expenditure_SA = as.numeric(Expenditure_SA),
    GDP_SA = as.numeric(GDP_SA)
  ) %>%


# Convert variables (except interest rate) to logs
  # Note multiplying by 10^6 for variables that are defined in £millions
  mutate(log_revenue = log(Revenue_SA *10^6),
         log_expenditure = log(Expenditure_SA *10^6),
         log_GDP = log(GDP_SA *10^6),
         log_deflator = log(Deflator),
         log_ERI = log(ERI),
         log_exports = log(Exports *10^6)) %>%
  mutate(
    dif_log_revenue = log_revenue - lag(log_revenue),
    dif_log_expenditure = log_expenditure - lag(log_expenditure),
    dif_log_GDP = log_GDP - lag(log_GDP),
    dif_log_deflator = log_deflator - lag(log_deflator),
```

```r
    dif_log_ERI = log_ERI - lag(log_ERI),
    dif_interest_rate = mean_SR_Rate - lag(mean_SR_Rate),
    dif_log_exports = log_exports - lag(log_exports)
  )


model_data2 <- data %>%
  dplyr::select(Year, CDID, log_expenditure, mean_SR_Rate, log_GDP, log_ERI, log_revenue,  l

# model_data <- data %>%
#   dplyr::select(Year, CDID, log_expenditure, mean_SR_Rate, log_exports, log_GDP, log_reven

# model_data <- data %>%
#   dplyr::select(Year, CDID, dif_log_expenditure, dif_interest_rate, dif_log_exports, dif_l

model_data <- data %>%
dplyr::select(Year, CDID, dif_log_expenditure, dif_interest_rate, dif_log_GDP, dif_log_ERI,
```

```r
# plot(seas(ts(data$Expenditure, start = min(data$Year), frequency = 4)))
plot(seas(ts(data$Revenue, start = min(data$Year), frequency = 4)))
```

```r
ggplot(data, aes(x = Unique_Period)) +
  geom_line(aes(y = Revenue, color = "Revenue"), size = 1) +
  geom_line(aes(y = Expenditure, color = "Expenditure"), size = 1) +
  labs(
    x = "Date ID",
    y = "Amount (? in millions)",
    title = "Revenue and Expenditure Over Time",
    color = "Legend"
  ) +
  scale_color_manual(values = c("Revenue" = "blue", "Expenditure" = "red")) +
```

```
  theme_minimal(base_size = 15) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title.x = element_text(face = "bold"),
    axis.title.y = element_text(face = "bold"),
    legend.position = "bottom",
    legend.title = element_text(face = "bold")
  )



ggplot(data, aes(x = Unique_Period)) +
  geom_line(aes(y = Revenue_SA, color = "Revenue"), size = 1) +
  geom_line(aes(y = Expenditure_SA, color = "Expenditure"), size = 1) +
  labs(
    x = "Date ID",
    y = "Amount (? in millions)",
    title = "Seasonally Adjusted Revenue and Expenditure Over Time",
    color = "Legend"
  ) +
  scale_color_manual(values = c("Revenue" = "blue", "Expenditure" = "red")) +
  theme_minimal(base_size = 15) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title.x = element_text(face = "bold"),
    axis.title.y = element_text(face = "bold"),
    legend.position = "bottom",
    legend.title = element_text(face = "bold")
  )



ggplot(data, aes(x = Unique_Period)) +
  geom_line(aes(y = Exports, color = "GDP"), size = 1) +
  labs(
    x = "Date ID",
    y = "Amount (? in millions)",
    title = "Exports Over Time",
    color = "Legend"
  ) +
  scale_color_manual(values = c("Exports" = "red")) +
  theme_minimal(base_size = 15) +
  theme(
```

```r
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title.x = element_text(face = "bold"),
    axis.title.y = element_text(face = "bold"),
    legend.position = "bottom",
    legend.title = element_text(face = "bold")
  )



ggplot(data, aes(x = Unique_Period)) +
  geom_line(aes(y = ERI, color = "ERI"), size = 1) +
  labs(
    x = "Date ID",
    y = "Amount (? in millions)",
    title = "Exchange Rate Index Over Time",
    color = "Legend"
  ) +
  scale_color_manual(values = c("ERI" = "red")) +
  theme_minimal(base_size = 15) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title.x = element_text(face = "bold"),
    axis.title.y = element_text(face = "bold"),
    legend.position = "bottom",
    legend.title = element_text(face = "bold")
  )



ggplot(data, aes(x = Unique_Period)) +
  geom_line(aes(y = GDP, color = "GDP"), size = 1) +
  labs(
    x = "Date ID",
    y = "Amount (? in millions)",
    title = "GDP Over Time",
    color = "Legend"
  ) +
  scale_color_manual(values = c("Exports" = "red")) +
  theme_minimal(base_size = 15) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title.x = element_text(face = "bold"),
```

```
    axis.title.y = element_text(face = "bold"),
    legend.position = "bottom",
    legend.title = element_text(face = "bold")
  )



# Define scaling factor based on range ratios
scale_factor2 <- max(data$Deflator, na.rm = TRUE) / max(data$mean_SR_Rate, na.rm = TRUE)

ggplot(data, aes(x = Unique_Period)) +
  geom_line(aes(y = Deflator, color = "Deflator"), size = 1) +
  geom_line(aes(y = mean_SR_Rate * scale_factor2, color = "Mean SR Rate"), size = 1, linetyp
  scale_y_continuous(
    name = "Deflator ",
    sec.axis = sec_axis(~ . / scale_factor2, name = "Mean SR Rate (%)")
  ) +
  labs(
    x = "Date ID",
    title = "Deflator and Mean SR Rate Over Time",
    color = "Legend"
  ) +
  scale_color_manual(values = c("Deflator" = "blue", "Mean SR Rate" = "Red")) +
  theme_minimal(base_size = 15) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title.x = element_text(face = "bold"),
    axis.title.y = element_text(face = "bold"),
    legend.position = "bottom",
    legend.title = element_text(face = "bold")
  )

ggplot(data, aes(x = Unique_Period)) +
  geom_line(aes(y = Deflator, color = "Deflator"), size = 1) +
  labs(
    x = "Date ID",
    y = "Amount (? in millions)",
    title = "Deflator Over Time",
    color = "Legend"
  ) +
  scale_color_manual(values = c("Deflator" = "blue", "Expenditure" = "red")) +
  theme_minimal(base_size = 15) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
```

```r
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title.x = element_text(face = "bold"),
    axis.title.y = element_text(face = "bold"),
    legend.position = "bottom",
    legend.title = element_text(face = "bold")
  )



clean_data <- na.omit(model_data)
clean_data2 <- na.omit(model_data2)

tmp <- clean_data[,-c(1,2)]

OptimalLag <-  VARselect(clean_data[,-c(1,2)], lag.max = 5, type = "const")
OptimalLag <-  VARselect(clean_data2[,-c(1,2)], lag.max = 5, type = "both")
OptimalLag$selection
# OptimalLag$criteria

# library(vars)

reduced_VAR <- VAR(clean_data[,-c(1,2)], p = 1, type = "const")
reduced_VAR2 <- VAR(clean_data2[,-c(1,2)], p = 4, type = "both")
# reduced_VAR <- VAR(clean_data[, -1], p = 4)

roots(reduced_VAR)
roots(reduced_VAR2)

# summary(reduced_VAR)

# Summary reports the roots of the polynomial.


# roots(reduced_VAR)





Amat <- matrix(c(1, 0, 0, 0, 0, 0,   # Recursive ordering
                 NA, 1, 0, 0, 0, 0,
                 NA, NA, 1, 0, 0, 0,
                 NA, NA, NA, 1, 0, 0,
                 NA, NA, NA, NA, 1, 0,
                 NA, NA, NA, NA, NA, 1),
```

```r
               nrow = 6, byrow = TRUE)



svar_model <- SVAR(reduced_VAR, Amat = Amat, estmethod = "direct")
svar_model2 <- SVAR(reduced_VAR2, Amat = Amat, estmethod = "direct")
# ?SVAR()

structural_shocks <- residuals(svar_model)



cov_matrix <- summary(reduced_VAR)$covres
chol(cov_matrix)



library(vars)

# Step 1: Extract covariance matrix from the reduced-form VAR
cov_matrix <- summary(reduced_VAR)$covres

# Step 2: Compute Cholesky decomposition
chol_matrix <- t(chol(cov_matrix))

# Step 3: Apply zero restrictions directly
# Example: Set specific elements to zero manually (adjust indices based on theory)
B_tilde <- chol_matrix
# B_tilde[2,1] <- 0  # Example: Restrict second variable's response to first shock
# B_tilde[3,1] <- 0  # Restrict third variable's response to first shock

# Step 4: Verify imposed restrictions
# print(B_tilde)

# Normalize B_tilde so the diagonal elements are 1
normalize_matrix <- function(B) {
  diag_values <- diag(B)  # Extract diagonal elements
  B_normalized <- B / diag_values  # Divide each column by its diagonal element
  return(B_normalized)
}

# Apply normalization
B_tilde_normalized <- normalize_matrix(B_tilde)

# Verify the transformation
```

```r
print(B_tilde_normalized)



# # Function to compute IRFs
# compute_irf <- function(A, B, horizon) {
#   k <- ncol(A)
#   IRF <- array(0, dim = c(k, k, horizon + 1))
#   IRF[, , 1] <- B  # Impact at time 0
#
#   for (h in 2:(horizon + 1)) {
#     IRF[, , h] <- A %*% IRF[, , h - 1]
#   }
#
#   return(IRF)
# }
#
# # Example execution:
# A_matrix <- matrix(c(0.5, 0.2, 0.1, 0.3, 0.7, -0.2, -0.1, 0.4, 0.6), 3, 3)  # Replace with
# IRFs <- compute_irf(A_matrix, B_tilde_normalized, horizon = 10)
# print(IRFs)



# Define the 5 dimensional lower triangular matrix, A

# Recover structural VAR using Cholesky decomposition

# Amat <- matrix(c(1, 0, 0, 0, 0,   # Recursive ordering
#                  NA, 1, 0, 0, 0,
#                  NA, NA, 1, 0, 0,
#                  NA, NA, NA, 1, 0,
#                  NA, NA, NA, NA, 1),
#                nrow = 5, byrow = TRUE)


Amat <- matrix(c(1, 0, 0, 0, 0, 0,   # Recursive ordering
                 NA, 1, 0, 0, 0, 0,
                 NA, NA, 1, 0, 0, 0,
                 NA, NA, NA, 1, 0, 0,
                 NA, NA, NA, NA, 1, 0,
                 NA, NA, NA, NA, NA, 1),
               nrow = 6, byrow = TRUE)
```

```
svar_model <- SVAR(reduced_VAR, Amat = Amat, estmethod = "direct")
svar_model2 <- SVAR(reduced_VAR2, Amat = Amat, estmethod = "direct")
# ?SVAR()

structural_shocks <- residuals(svar_model)

svar_model

irf_result <- irf(svar_model, n.ahead = 10, ci = 0.68, boot = 5000, cumulative = TRUE)  # Fo
irf_result2 <- irf(svar_model2, n.ahead = 10, ci = 0.68, boot = 5000, cumulative = TRUE)  #
# irf_result <- irf(svar_model, n.ahead = 10, ci = 0.68, boot = 5000, cumulative = TRUE)  #
# plot(irf_result)
# Visualize IRFs

FEVD_result <- fevd(svar_model, n.ahead = 10)  # Forecast horizons
FEVD_result2 <- fevd(svar_model2, n.ahead = 10)  # Forecast horizons
plot(FEVD_result)
plot(FEVD_result2)

plot_irf_with_ci <- function(IRF_name) {


  # Extract response and confidence intervals
  irf_data  <- as.data.frame(IRF_name$irf)
  irf_lower <- as.data.frame(IRF_name$Lower)
  irf_upper <- as.data.frame(IRF_name$Upper)

  # Create time index
  irf_data$Time  <- seq_len(nrow(irf_data))
  irf_lower$Time <- irf_data$Time
  irf_upper$Time <- irf_data$Time

  # Reshape to long format
  irf_long   <- pivot_longer(irf_data, cols = -Time, names_to = "Variable", values_to = "IRF
  lower_long <- pivot_longer(irf_lower, cols = -Time, names_to = "Variable", values_to = "Lo
  upper_long <- pivot_longer(irf_upper, cols = -Time, names_to = "Variable", values_to = "Up

  # Merge and label
  irf_combined <- irf_long %>%
    left_join(lower_long, by = c("Time", "Variable")) %>%
    left_join(upper_long, by = c("Time", "Variable")) %>%
    mutate(
      Shock = sub("\\..*", "", Variable),
      Affected_Var = sub(".*\\.", "", Variable)
    )
```

```r
  # Plot IRFs by shock
  shock_names <- unique(irf_combined$Shock)

  for (shock in shock_names) {
    p <- irf_combined %>%
      filter(Shock == shock) %>%
      ggplot(aes(x = Time, y = IRF)) +
      geom_line(size = 1.2, color = "#1f77b4") +
      geom_line(aes(y = Lower), linetype = "dashed", color = "#7f7f7f", size = 0.9) +
      geom_line(aes(y = Upper), linetype = "dashed", color = "#7f7f7f", size = 0.9) +
      geom_hline(yintercept = 0, color = "black", size = 1.1) +
      facet_wrap(~ Affected_Var, scales = "free_y") +
      theme_minimal(base_family = "serif") +
      theme(
        plot.title = element_text(size = 16, face = "bold"),
        axis.title.x = element_text(size = 14),
        axis.title.y = element_text(size = 14),
        axis.text = element_text(size = 12),
        panel.grid.major = element_line(color = "gray85"),
        panel.grid.minor = element_blank(),
        strip.text = element_text(size = 14, face = "bold")
      ) +
      labs(
        title = paste("Impulse Response for Shock:", shock),
        x = "Time",
        y = "Response"
      ) +
      scale_y_continuous(expand = expansion(mult = c(0, 0.05)))

    print(p)
  }
}

plot_irf_with_ci(irf_result)
plot_irf_with_ci(irf_result2)



FEVD_result



# structural_shocks <- residuals(svar_model)
# irf_result <- irf(svar_model, n.ahead = 10)  # Forecast horizons
# plot(irf_result)  # Visualize IRFs
```

```r
normality.test(reduced_VAR)




names(model_data)

mean(model_data$dif_log_expenditure, na.rm = TRUE)
mean(model_data$dif_interest_rate, na.rm = TRUE)
mean(model_data$dif_log_exports, na.rm = TRUE)
mean(model_data$dif_log_GDP, na.rm = TRUE)
mean(model_data$dif_log_revenue, na.rm = TRUE)
mean(model_data$dif_deflator_rate, na.rm = TRUE)

run_var_analysis <- function(model_data) {

  # Extract suffix from input name
  input_name <- deparse(substitute(model_data))
  suffix <- sub("^[^_]+_", "", input_name)
  suffix <- ifelse(nchar(suffix) == 0, "data", suffix)

  # Split the data into pre- and post-2008 samples
clean_data_a <- model_data %>%
  filter(Year < 2008 & complete.cases(.))
clean_data_b <- model_data %>% filter(!is.na(Year) & Year >= 2008)

  assign(paste0("clean_data_a_", suffix), clean_data_a, envir = .GlobalEnv)
  assign(paste0("clean_data_b_", suffix), clean_data_b, envir = .GlobalEnv)

  # Lag selection
  lag_a <- VARselect(clean_data_a[,-c(1,2)], lag.max = 5, type = "const")$selection
  lag_b <- VARselect(clean_data_b[,-c(1,2)], lag.max = 5, type = "const")$selection
  print(list(OptimalLag_Pre2008 = lag_a, OptimalLag_Post2008 = lag_b))

  # Estimate reduced-form VARs
  var_a <- VAR(clean_data_a[,-c(1,2)], p = 4, type = "const")
  var_b <- VAR(clean_data_b[,-c(1,2)], p = 4, type = "const")

  assign(paste0("reduced_VAR_a_", suffix), var_a, envir = .GlobalEnv)
  assign(paste0("reduced_VAR_b_", suffix), var_b, envir = .GlobalEnv)

  # Check stability
```

```r
  print(roots(var_a))
  print(roots(var_b))

  # Recursive identification matrix
  k <- ncol(clean_data_a[,-c(1,2)])
  Amat <- diag(1, k)
  Amat[upper.tri(Amat)] <- NA

  # Estimate SVARs
  svar_a <- SVAR(var_a, Amat = Amat, estmethod = "direct")
  svar_b <- SVAR(var_b, Amat = Amat, estmethod = "direct")

  assign(paste0("svar_model_a_", suffix), svar_a, envir = .GlobalEnv)
  assign(paste0("svar_model_b_", suffix), svar_b, envir = .GlobalEnv)

  # IRFs
  irf_a <- irf(svar_a, n.ahead = 10, ci = 0.68, boot = 5000, cumulative = TRUE)
  irf_b <- irf(svar_b, n.ahead = 10, ci = 0.68, boot = 5000, cumulative = TRUE)

  assign(paste0("irf_result_a_", suffix), irf_a, envir = .GlobalEnv)
  assign(paste0("irf_result_b_", suffix), irf_b, envir = .GlobalEnv)

  # FEVDs
  fevd_a <- fevd(svar_a, n.ahead = 10)
  fevd_b <- fevd(svar_b, n.ahead = 10)

  assign(paste0("FEVD_result_a_", suffix), fevd_a, envir = .GlobalEnv)
  assign(paste0("FEVD_result_b_", suffix), fevd_b, envir = .GlobalEnv)

  # Plot FEVDs
  plot(fevd_a)
  plot(fevd_b)

  # irf_result_a <- irf(svar_a, n.ahead = 10, ci = 0.68, boot = 5000, cumulative = TRUE)  #
  # irf_result_b <- irf(svar_b, n.ahead = 10, ci = 0.68, boot = 5000, cumulative = TRUE)  #


}
run_var_analysis(model_data)

run_var_analysis(model_data2)


# plot_irf_with_ci(irf_result2_data)
```

```r
# Apply processing and plotting functions


plot_irf_with_ci(irf_result_a_data)
plot_irf_with_ci(irf_result_a_data2)

# normality.test(reduced_VAR1)

# irf_long1 <- process_irf(irf_result1_data)

# plot_irf(irf_long1)

# roots(reduced_VAR1)




# # Function to process IRF data for plotting
# process_irf <- function(irf_result) {
#   # Convert IRF results to data frames and add time index
#   irf_long <- irf_result$irf %>%
#     as.data.frame() %>%
#     mutate(Time = seq_len(nrow(.))) %>%
#     pivot_longer(cols = -Time, names_to = "Variable", values_to = "IRF")
#
#   lower_long <- irf_result$Lower %>%
#     as.data.frame() %>%
#     mutate(Time = seq_len(nrow(.))) %>%
#     pivot_longer(cols = -Time, names_to = "Variable", values_to = "Lower")
#
#   upper_long <- irf_result$Upper %>%
#     as.data.frame() %>%
#     mutate(Time = seq_len(nrow(.))) %>%
#     pivot_longer(cols = -Time, names_to = "Variable", values_to = "Upper")
#
#   # Merge confidence bounds and extract variable components
#   irf_long <- irf_long %>%
#     left_join(lower_long, by = c("Time", "Variable")) %>%
#     left_join(upper_long, by = c("Time", "Variable")) %>%
#     mutate(Shock = sub("\\..*", "", Variable),  # Extract shock name
#            Affected_Var = sub(".*\\.", "", Variable))  # Extract affected variable
#
#   return(irf_long)
# }
#
# # Function to plot IRFs with facets
```

```r
# plot_irf <- function(irf_long) {
#   shock_names <- unique(irf_long$Shock)
#
#   # Generate plots for each shock
#   for (shock in shock_names) {
#     p <- ggplot(irf_long %>% filter(Shock == shock), aes(x = Time, y = IRF)) +
#       geom_line(size = 1.2, color = "#1f77b4") +  # Darker blue for better contrast
#       geom_line(aes(y = Lower), linetype = "dashed", color = "#7f7f7f", size = 0.9) +  # S
#       geom_line(aes(y = Upper), linetype = "dashed", color = "#7f7f7f", size = 0.9) +
#       geom_hline(yintercept = 0, color = "black", size = 1.1) +  # Stronger baseline
#
#       facet_wrap(~ Affected_Var, scales = "free_y") +
#       theme_minimal(base_family = "serif") +  # Professional font
#
#       theme(
#         plot.title = element_text(size = 16, face = "bold"),  # Larger title
#         axis.title.x = element_text(size = 14),  # Larger axis labels
#         axis.title.y = element_text(size = 14),
#         axis.text = element_text(size = 12),  # Bigger tick labels
#         panel.grid.major = element_line(color = "gray85"),  # Softer grid lines
#         panel.grid.minor = element_blank(),  # Hide minor grid lines
#         strip.text = element_text(size = 14, face = "bold")  # Improve facet labels
#       ) +
#
#       labs(title = paste("Impulse Response for Shock:", shock),
#            x = "Time",
#            y = "Response") +
#
#       scale_y_continuous(expand = expansion(mult = c(0, 0.05)))  # Avoid clipping near ze
#
#     print(p)  # Render each plot separately
#   }
#
#
# }


plot_irf_with_ci(irf_result_b_data)
plot_irf_with_ci(irf_result_b_data2)


# normality.test(reduced_VAR2)
#
# irf_long2 <- process_irf(irf_result2)
#
```

```
#
# plot_irf(irf_long2)
#
# roots(reduced_VAR2)
```

References

[1] O.J. Blanchard and D. Leigh. "Growth forecast errors and fiscal multipliers". In: *American Economic Review* 103.3 (2013), pp. 117–120.

[2] E. Ilzetzki, E.G. Mendoza, and C.A. Végh. "How big (small?) are fiscal multipliers?" In: *Journal of Monetary Economics* 60.2 (2013), pp. 239–254.

[3] D. Sutherland, P. Hoeller, and R. Merola. "Fiscal consolidation: How much, how fast and by what means?" In: (2012).

[4] T. Warmedinger, C.D. Checherita-Westphal, and P.H. De Cos. "Fiscal multipliers and beyond". In: *ECB Occasional Paper* 162 (2015).