```
clear; clc;
rng(123456);
% ========================== PC2: Exe2 ===============================
% ====================================================================
% Import data
data      = readtable('RRdata.xlsx');  % function: readtable
GDPgr     = data.PCGDP1;               % GDP growth (PCGDP1)
Tax       = data.EXOGENRRATIO;         % Tax shock (EXOGENRRATIO)
timeraw      = (datetime(1947,01,01):calquarters(1):datetime(2007,12,01))';
%% ------------------------ part 1.a) ------------------------------
% Estimate the model: GDPgr= a + sum^{M=12}_{i=0} b_i Tax_{t-i} + u_t;
% In the matrix form
%                     Y_t = X_t B + u_t
% where:
%          Y_t = GDPgr
%          X_t = [1 X_t X_{t-1} .. X_{t-M}]
%          B = [a b_i]'
% ----------------------------- Set up -----------------------------
M        = 12; % lag length
Xlags    = lagmatrix(Tax,0:M);                                 % create lag
matrix: lagmatrix
X        = Xlags(M+1:end,:);                      % sufficient observations of X
Y        = GDPgr(M+1:end);                        % sufficient observations of Y
time     = timeraw(M+1:end);
hor      = 0:1:M;                                 % number of response horizon
K        = size(Xlags,2)+1;                       % number of estimators
T        = size(Xlags,1);                         % number of observations in full
samples
% --------------------------- Estimation ------------------------------
mdlest = fitlm(X,Y);                                     % estimate model using
fitlm
disp(mdlest)                                      % display regression output
```

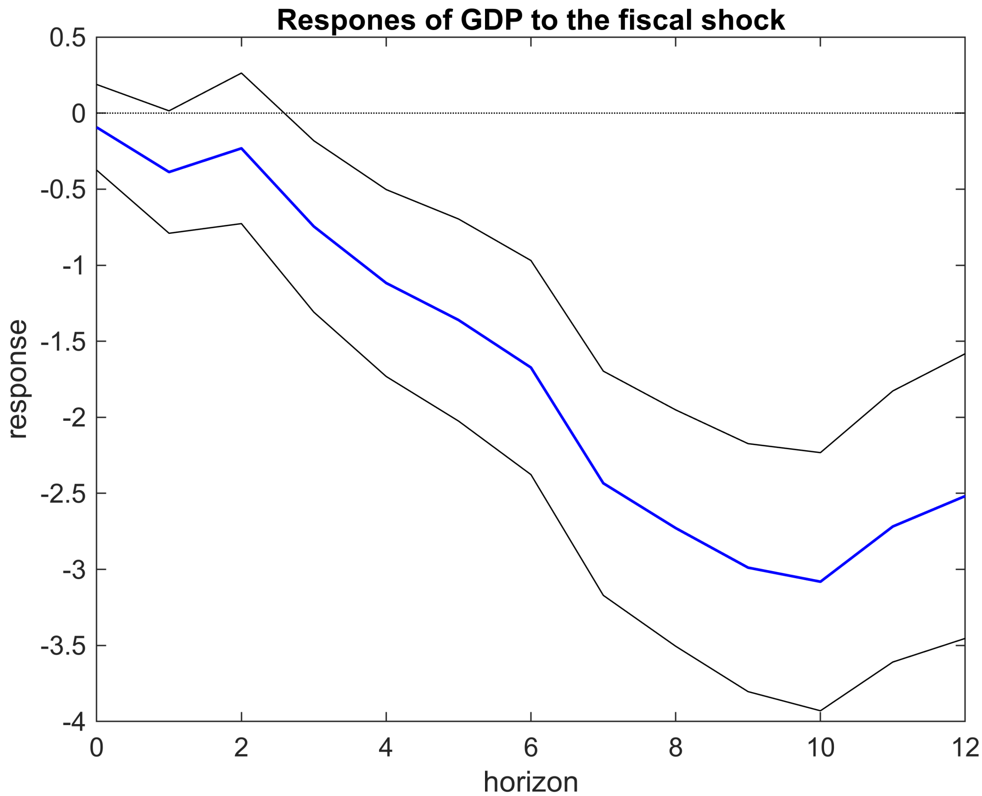
```
Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 + x12 + x13

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
(Intercept)	0.79135	0.066308	11.934	1.3455e-25
x1	-0.093414	0.29063	-0.32142	0.7482
x2	-0.29417	0.29048	-1.0127	0.31232
x3	0.15591	0.29003	0.53755	0.59144
x4	-0.51337	0.29023	-1.7688	0.078327
x5	-0.37203	0.29098	-1.2786	0.20242
x6	-0.24342	0.29056	-0.83777	0.40308
x7	-0.31234	0.29068	-1.0745	0.28379
x8	-0.76069	0.25272	-3.0101	0.0029197
x9	-0.29469	0.25313	-1.1642	0.24564
x10	-0.26009	0.25277	-1.029	0.30463
x11	-0.09251	0.25247	-0.36641	0.71441
x12	0.36308	0.25262	1.4373	0.15208
x13	0.20004	0.25588	0.78177	0.4352

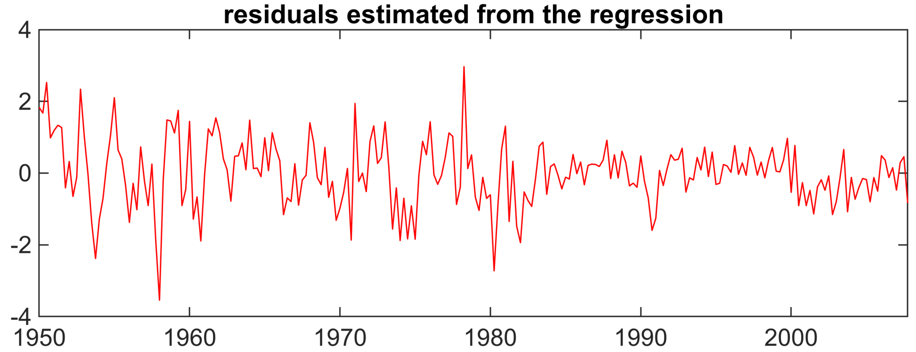
```
Number of observations: 232, Error degrees of freedom: 218
Root Mean Squared Error: 0.946
R-squared: 0.102,  Adjusted R-Squared: 0.0488
F-statistic vs. constant model: 1.91, p-value = 0.0302
```

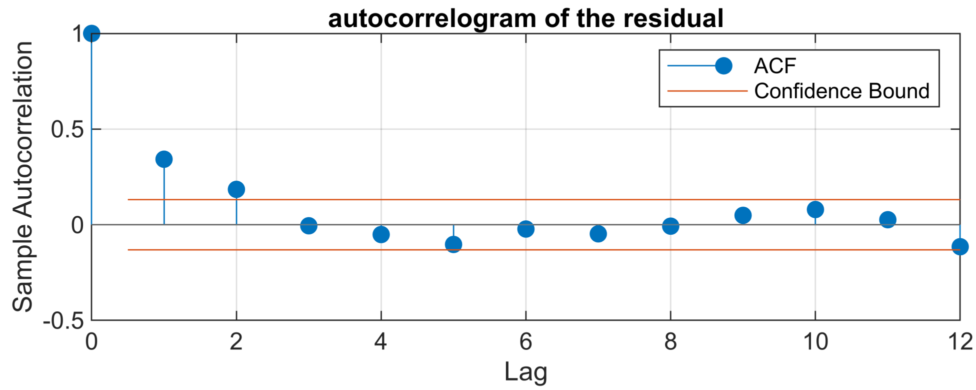
```
% % ===== store the estimated values
% Bhat    = mdlest.Coefficients.Estimate;            % store the estimators
% res     = mdlest.Residuals.Raw;                    % store the resiudals
% VarBhat = mdlest.CoefficientCovariance;

%% =========== OLS estimator
% sufficient number of observations for estimation
T1 = T-M;
Xreg = [ones(T1,1) X];
Bhat =(Xreg'*Xreg)\(Xreg'*Y);  % OLS estimates
res = Y-Xreg*Bhat;
Sigma_u = (1/T1)*res'*res;
VarBhat = inv(Xreg'*Xreg)*Sigma_u;

% ===== cumulated effects
beta    = Bhat(2:end);                              % beta=[b0 b1 ...b12]'
resp    = cumsum(beta) ;                                      % cumsum
% use bootstrap to calculate the standard error estimation for
% impulse responses (cumulative effects)
% ======================== boostrap =====================
ndraws=10000;
P = chol(VarBhat,'lower');  % Cholesky decomposition.
resp_bstr=zeros(ndraws,M+1);
% start boostrappingg
for j=1:ndraws
    coeff_bstr=Bhat+P*mvnrnd(0,1,K); % draw beta from the normal distribution
    b_bstr=coeff_bstr(2:end);         % estimated coefficients of tax shocks
    bsum=cumsum(b_bstr);              % the cumulated effects
    resp_bstr(j,:)=bsum';
end
%=========================== end boostrap ================
% mean and estimated standard error
resp_mean = mean(resp_bstr,1)';            % mean
resp_std  = std(resp_bstr,1)';             % std
% t-value
gam_tval  = resp./resp_std;
% lower and upper bound of 68% CI
resp_lb   = resp+tinv(0.16,T-M)*resp_std;  % lower bound
resp_ub   = resp+tinv(0.84,T-M)*resp_std;  % upper bound
% ===== plot
figure
plot(hor,resp,'b','LineWidth',1); hold on;
plot(hor,resp_lb,'-k'); hold on;
plot(hor,resp_ub,'-k'); hold on;
```

```
plot(hor,zeros(M+1,1),':k');
xlim([0 M]);
xlabel('horizon');
ylabel('response');
title('Respones of GDP to the fiscal shock');
```




```
%% ------------------------ Part 1b:  Autocorrelation -------------------
% ===== plot autocorrelogram: autocorr
figure
subplot(2,1,1)
plot(time,res,'-r');
title('residuals estimated from the regression')
subplot(2,1,2)
autocorr(res,12)
ylim([-0.5 1])
title('autocorrelogram of the residual')
```






```
% ===== Ljung-Box Q-test for residual autocorrelation: lbqtest
[~,pval_lbq,stat_lbq,cval] = lbqtest(res,"Lags",1:12);
tab_lbq = table(pval_lbq',stat_lbq',cval');
disp(tab_lbq)
```

Var1	Var2	Var3
1.5201e-07	27.564	3.8415
1.8383e-08	35.624	5.9915
8.963e-08	35.631	7.8147
2.5616e-07	36.257	9.4877
2.6052e-07	38.8	11.07
7.4136e-07	38.922	12.592
1.5917e-06	39.467	14.067
4.0021e-06	39.481	15.507
7.3948e-06	40.065	16.919
8.7972e-06	41.61	18.307
1.769e-05	41.779	19.675
1.0137e-05	45.042	21.026

```
%% --------------- Part 1c: Newey West standard error --------------------
% ===== Newey-West covariance matrix estimation: hac
q = round(T^(1/4));
covNW = hac(X,Y,"Bandwidth",q+1);
```

```
Estimator type: HAC
Estimation method: BT
Bandwidth: 5.0000
```

```
Whitening order: 0
Effective sample size: 232
Small sample correction: on

Coefficient Covariances:

       |  Const      x1       x2       x3       x4       x5       x6       x7       x8       x9      x10      x11
-------------------------------------------------------------------------------------------------------------------
 Const |  0.0061  -0.0008   0.0016   0.0005  -0.0004   0.0050   0.0011   0.0039   0.0015  -0.0010  -0.0027  -0.0010
 x1    | -0.0008   0.1140   0.0163   0.0375  -0.0091  -0.0478  -0.0029  -0.0187  -0.0242  -0.0128  -0.0039  -0.0024
 x2    |  0.0016   0.0163   0.0584   0.0200   0.0003  -0.0057  -0.0098  -0.0016  -0.0083   0.0006  -0.0031  -0.0007
 x3    |  0.0005   0.0375   0.0200   0.0404   0.0110  -0.0078   0.0010  -0.0104  -0.0184  -0.0183  -0.0201  -0.0069
 x4    | -0.0004  -0.0091   0.0003   0.0110   0.0806   0.0250   0.0086  -0.0213  -0.0128  -0.0164  -0.0119   0.0036
 x5    |  0.0050  -0.0478  -0.0057  -0.0078   0.0250   0.0556   0.0078   0.0002  -0.0011  -0.0127  -0.0097  -0.0048
 x6    |  0.0011  -0.0029  -0.0098   0.0010   0.0086   0.0078   0.0508   0.0236  -0.0015  -0.0154  -0.0223  -0.0063
 x7    |  0.0039  -0.0187  -0.0016  -0.0104  -0.0213   0.0002   0.0236   0.0643   0.0098   0.0055  -0.0183  -0.0110
 x8    |  0.0015  -0.0242  -0.0083  -0.0184  -0.0128  -0.0011  -0.0015   0.0098   0.0843   0.0468   0.0517   0.0082
 x9    | -0.0010  -0.0128   0.0006  -0.0183  -0.0164  -0.0127  -0.0154   0.0055   0.0468   0.0804   0.0667   0.0210
 x10   | -0.0027  -0.0039  -0.0031  -0.0201  -0.0119  -0.0097  -0.0223  -0.0183   0.0517   0.0667   0.1298   0.0372
 x11   | -0.0010  -0.0024  -0.0007  -0.0069   0.0036  -0.0048  -0.0063  -0.0110   0.0082   0.0210   0.0372   0.0378
 x12   | -0.0007   0.0140  -0.0008   0.0050   0.0011  -0.0048  -0.0027  -0.0073  -0.0082   0.0008   0.0175   0.0160
 x13   | -0.0003   0.0222   0.0047   0.0093   0.0047  -0.0060  -0.0032  -0.0164  -0.0077  -0.0072   0.0162   0.0205
```

```matlab
% =========================== boostrap ==================================
ndraws=10000;
P=chol(covNW,'lower');  % Cholesky decomposition.
resp_bstr=zeros(ndraws,M+1);
% start boostrapping
for j=1:ndraws
    ABhat_bstr=Bhat+P*mvnrnd(0,1,K); % draw beta from the normal distribution
    b_bstr=ABhat_bstr(2:end);         % estimated coefficients of tax shocks
    bsum=cumsum(b_bstr);              % the cumulated effects
    resp_bstr(j,:)=bsum';
end
% ===== Mean and Estimated standard error for the responses
respmean     = mean(resp_bstr,1)';
respseNW     = std(resp_bstr,1)';

% ======================== End boostrap =============================
% ===== Newey-West t-value
gam_NWtval=resp./respseNW;
% construct table
results3=array2table([hor', resp, gam_tval, gam_NWtval],"VariableNames",...
    ["horizon","responses","conventional t-value","Newey-West t-value"]);
disp(results3)
```

horizon	responses	conventional t-value	Newey-West t-value
0	-0.093414	-0.33056	-0.27603
1	-0.38759	-0.9597	-0.84829
2	-0.23168	-0.46655	-0.38482
3	-0.74505	-1.3178	-1.1217
4	-1.1171	-1.812	-1.7248
5	-1.3605	-2.0411	-1.9741
6	-1.6728	-2.369	-2.3988

7	-2.4335	-3.2901	-3.5598
8	-2.7282	-3.4996	-3.8181
9	-2.9883	-3.6532	-3.5721
10	-3.0808	-3.618	-3.4203
11	-2.7177	-3.0393	-2.8405
12	-2.5177	-2.6811	-2.403