In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
import itertools
```

Part A

Pre process

In [2]:
```
df=pd.read_csv('C:/Users/14857/Desktop/时序分析/final project(1)/final project/TS project.csv')
df.head()
```

Out[2]:

	Time	Year	Qualterly % Change of Inflation (qualterly)	Real GDP Change	Nominal GDP Change	Nikkei 225 Index	Quarterly Nikkei 225 Index Change	Monetary Base	Monetary Base (SA)	Quarterly Change MB	% Quarterly Change- MB (SA)	Income Inequality (disposable income)	Empl
0	1	1994	NaN	1.2	1.0	19474.75787	NaN	417272.6667	413229.0000	NaN	NaN	0.285403	
1	2	1994	NaN	-0.6	-0.2	20397.33508	0.047373	413218.3333	417997.6667	-0.009716	0.011540	0.243823	
2	3	1994	NaN	1.2	0.8	20369.57046	-0.001361	422350.6667	425975.6667	0.022101	0.019086	0.251858	
3	4	1994	NaN	-0.4	-0.3	19473.21672	-0.044005	431003.6667	427084.6667	0.020488	0.002603	0.208498	
4	5	1995	-0.62	1.1	0.7	17756.83164	-0.088141	435148.0000	430382.6667	0.009616	0.007722	0.277298	

In [3]:
```
#Name the columns
inflation_col="Qualterly % Change of Inflation (qualterly)"
gdp_change_col="Real GDP Change"
index_col="Nikkei 225 Index "
index_change_col="Quarterly Nikkei 225 Index Change"
mb_col="Monetary Base (SA)"
mb_change_col="% Quarterly Change- MB (SA)"
income_inequal_col="Income Inequality (disposable income)"
unemployment_rate_col="Unemployment rate"
earthquake_col="earthquake"
tax_col="tax"

#delete the Nah rows
df=df.dropna(axis=0,how='any')
print(len(df))
#Gen the unemployment rate
df['Unemployment rate']=100-df['Employment rate']
```

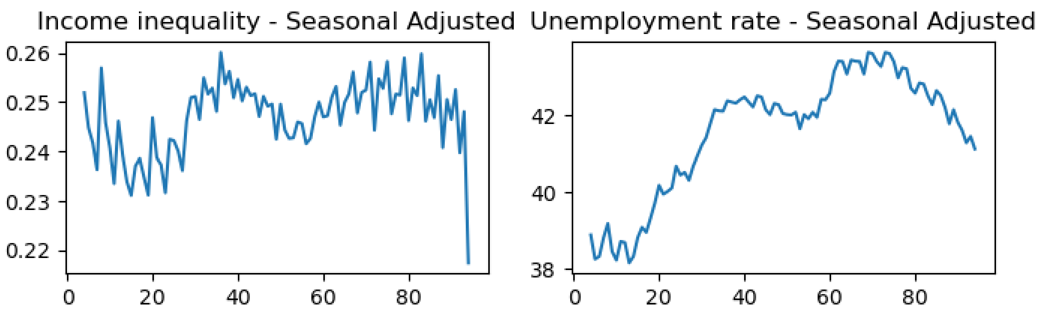
91

Plot the series.

In [4]:
```python
cols=list(df.columns)
cols.remove("Employment rate")
fig,axs=plt.subplots(5,3,figsize=(30,20))
for i,col in enumerate(cols):
    axs[i//3,i%3].plot(df[col])
    axs[i//3,i%3].set_title(col)
plt.show()
```

Adjust the series which have seasonality.

In [5]:
```python
from statsmodels.tsa.seasonal import seasonal_decompose

#decompose the seasonality

sd=seasonal_decompose(df[income_inequal_col],period=4,model="additive")
income_inequal_sa_col="Income inequality - Seasonal Adjusted"
df[income_inequal_sa_col]=df[income_inequal_col]-sd.seasonal

sd=seasonal_decompose(df[unemployment_rate_col],period=4,model="additive")
unemployment_sa_col="Unemployment rate - Seasonal Adjusted"
df[unemployment_sa_col]=df[unemployment_rate_col]-sd.seasonal

#plot the adjusted series
fig,axs=plt.subplots(1,2,figsize=(8,2))
axs[0].plot(df[income_inequal_sa_col])
axs[0].set_title(income_inequal_sa_col)
axs[1].plot(df[unemployment_sa_col])
axs[1].set_title(unemployment_sa_col)
plt.show()
```



Test the time trend and do the ADF test.

The Series which will be used in later.
inflation_col; gdp_change_col; index_col; index_change_col; mb_col; mb_change_col; income_inequal_sa_col; unemployment_sa_col; earthquake_col; tax_col;

In [6]:
```
from statsmodels.tsa.stattools import adfuller
from scipy.stats import linregress

target_cols=[inflation_col,gdp_change_col,index_col,index_change_col,mb_col,mb_change_col,income_inequal_sa_col,\
             unemployment_sa_col,earthquake_col,tax_col]
len(target_cols)

adf_tab=pd.DataFrame(columns=['Name','tau','p-value','5%','Lags','Obs','Trend'])
```

In [7]:
```python
from scipy.stats import linregress
from statsmodels.tsa.stattools import adfuller

def adf_test(data, adf_tab, name):
    _, _, _, p_value, _ = linregress(df["Time"], data)
    mode = "ct" if p_value < 0.05 else "c"
    print("---------------------------")
    print(mode)
    result = adfuller(data, regression=mode)
    print('ADF Statistic: %f' % result[0])
    print('p-value: %f' % result[1])
    print('Critical Values:')
    for key, value in result[4].items():
        print('\t%s: %.3f' % (key, value))
    trend = 'No' if mode == 'c' else 'Yes'
    stationary = 'No' if result[1] > 0.05 else 'Yes'

    # Create a new DataFrame for this row and concatenate it with adf_tab
    new_row = pd.DataFrame({
        'Name': [name],
        'tau': [round(result[0], 2)],
        'p-Value': [round(result[1], 2)],
        '5%': [round(result[4]['5%'], 2)],
        'Lags':[result[2]],
        'Obs': [result[3]],
        'Trend': [trend],
        'Stationary': [stationary]
    })
    adf_tab = pd.concat([adf_tab, new_row], ignore_index=True)
    return adf_tab

# Assuming df and target_cols are defined earlier
adf_tab = pd.DataFrame()  # Initialize DataFrame to store results
for col in target_cols:
    adf_tab = adf_test(df[col], adf_tab, col)


display(adf_tab.set_index('Name'))
```

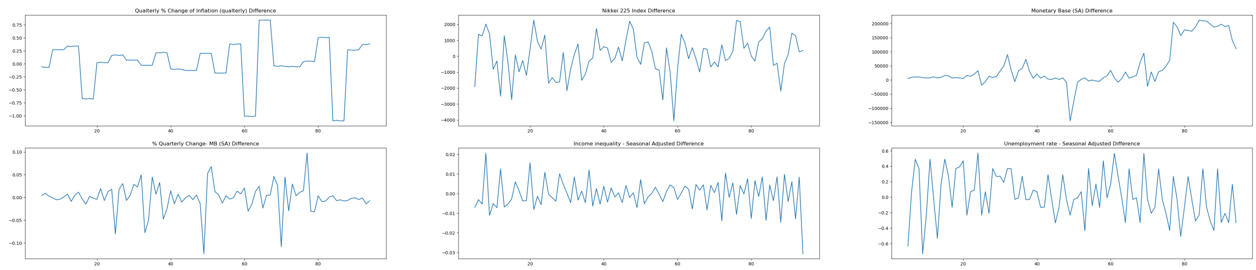
```
----------------------------
c
ADF Statistic: -3.530350
p-value: 0.007241
Critical Values:
        1%: -3.514
        5%: -2.898
        10%: -2.586
----------------------------
c
ADF Statistic: -5.807464
p-value: 0.000000
Critical Values:
        1%: -3.507
        5%: -2.895
        10%: -2.585
----------------------------
c
ADF Statistic: -1.843177
p-value: 0.359246
Critical Values:
        1%: -3.506
        5%: -2.895
        10%: -2.584
----------------------------
c
ADF Statistic: -6.690223
p-value: 0.000000
Critical Values:
        1%: -3.505
        5%: -2.894
        10%: -2.584
----------------------------
ct
ADF Statistic: -0.006960
p-value: 0.994162
Critical Values:
        1%: -4.064
        5%: -3.461
        10%: -3.157
----------------------------
ct
ADF Statistic: -4.321440
p-value: 0.002913
Critical Values:
        1%: -4.063
        5%: -3.460
        10%: -3.156
----------------------------
ct
ADF Statistic: -1.478231
p-value: 0.836467
Critical Values:
        1%: -4.077
        5%: -3.467
        10%: -3.160
----------------------------
ct
ADF Statistic: -0.283630
p-value: 0.989885
Critical Values:
        1%: -4.071
        5%: -3.464
        10%: -3.158
----------------------------
c
ADF Statistic: -4.692389
p-value: 0.000087
Critical Values:
        1%: -3.508
        5%: -2.895
        10%: -2.585
----------------------------
c
ADF Statistic: -9.486833
p-value: 0.000000
Critical Values:
        1%: -3.505
        5%: -2.894
        10%: -2.584
```

Name	tau	p-Value	5%	Lags	Obs	Trend	Stationary
Qualterly % Change of Inflation (qualterly)	-3.53	0.01	-2.90	9	81	No	Yes
Real GDP Change	-5.81	0.00	-2.89	2	88	No	Yes
Nikkei 225 Index	-1.84	0.36	-2.89	1	89	No	No
Quarterly Nikkei 225 Index Change	-6.69	0.00	-2.89	0	90	No	Yes
Monetary Base (SA)	-0.01	0.99	-3.46	1	89	Yes	No
% Quarterly Change- MB (SA)	-4.32	0.00	-3.46	0	90	Yes	Yes
Income inequality - Seasonal Adjusted	-1.48	0.84	-3.47	10	80	Yes	No
Unemployment rate - Seasonal Adjusted	-0.28	0.99	-3.46	6	84	Yes	No
earthquake	-4.69	0.00	-2.90	3	87	No	Yes
tax	-9.49	0.00	-2.89	0	90	No	Yes

Do the first order difference.

For the non-stationary series and series which are stationary but have time trend, we do first difference and for the interpretability, we just do first difference without apply higher difference.

In [8]:
```
inflation_diff_col="Qualterly % Change of Inflation (qualterly) Difference"
index_diff_col="Nikkei 225 Index Difference"
mb_diff_col="Monetary Base (SA) Difference"
mb_change_diff_col="% Quarterly Change- MB (SA) Difference"
income_inequal_diff_col="Income inequality - Seasonal Adjusted Difference"
unemployment_diff_col="Unemployment rate - Seasonal Adjusted Difference"

df[inflation_diff_col]=df[inflation_col].diff()
df[index_diff_col]=df[index_col].diff()
df[mb_diff_col]=df[mb_col].diff()
df[mb_change_diff_col]=df[mb_change_col].diff()
df[income_inequal_diff_col]=df[income_inequal_sa_col].diff()
df[unemployment_diff_col]=df[unemployment_sa_col].diff()

cols=[inflation_diff_col,index_diff_col,mb_diff_col,mb_change_diff_col,income_inequal_diff_col,unemployment_diff_col]
fig,axs=plt.subplots(2,3,figsize=(50,10))
for i,col in enumerate(cols):
    axs[i//3,i%3].plot(df[col])
    axs[i//3,i%3].set_title(col)
plt.show()
```



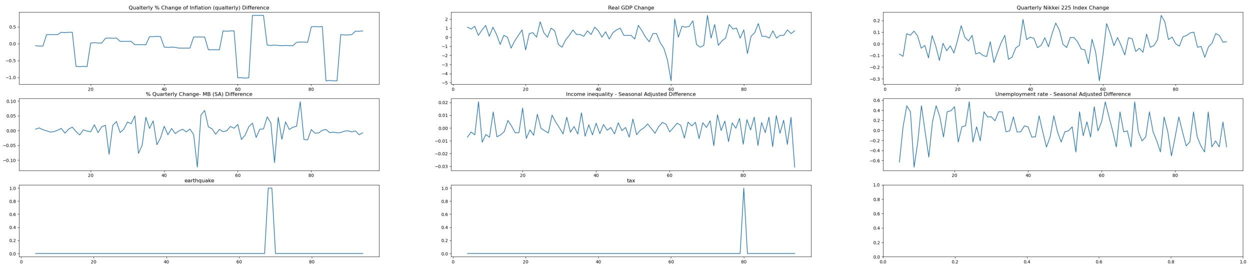
Use AIC/BIC to decide p in VAR(p).

Since mb_change_diff_col has the same information as mb_change_diff_col and index_change_col as index_diff_col, we just use mb_change_diff_col and index_change_col. And We plot the series used in VAR.

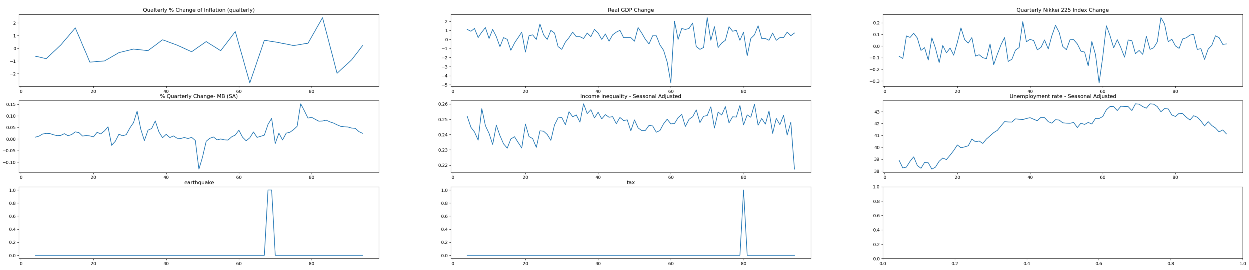
In [9]:
```
var_data1 = df[[inflation_diff_col,gdp_change_col,index_change_col,mb_change_diff_col,income_inequal_diff_col,\
                unemployment_diff_col,earthquake_col,tax_col]]
var_data1 = var_data1.dropna(axis=0, how="any")
var_data2 = df[[inflation_col,gdp_change_col,index_change_col,mb_change_col,income_inequal_sa_col,unemployment_sa_col,\
                earthquake_col,tax_col]]
var_data2 = var_data2.dropna(axis=0, how="any")
```

VAR_1 use the stationary series which with first order differnece process. And VAR_2 use the series with out differnece for it will be more easy to explain.

In [10]:
```
cols=[inflation_diff_col,gdp_change_col,index_change_col,mb_change_diff_col,income_inequal_diff_col,unemployment_diff_col,
fig,axs=plt.subplots(3,3,figsize=(50,10))
for i,col in enumerate(cols):
    axs[i//3,i%3].plot(df[col])
    axs[i//3,i%3].set_title(col)
plt.show()
```



In [11]:
```
cols=[inflation_col,gdp_change_col,index_change_col,mb_change_col,income_inequal_sa_col,unemployment_sa_col,earthquake_co
fig,axs=plt.subplots(3,3,figsize=(50,10))
for i,col in enumerate(cols):
    axs[i//3,i%3].plot(df[col])
    axs[i//3,i%3].set_title(col)
plt.show()
```



The Model selection process.

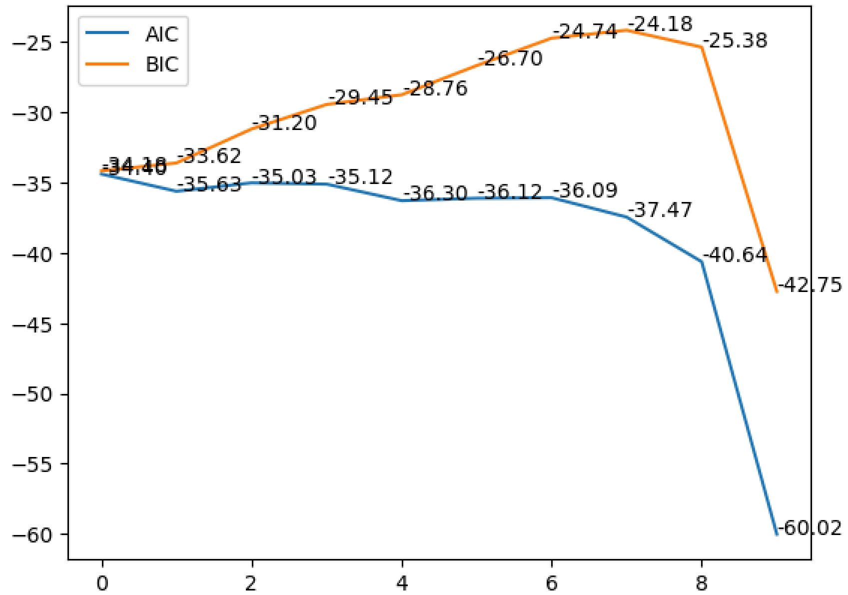
In [12]:

```
from statsmodels.tsa.api import VAR

#Construct a model selection method to choose the optimal AIC or BIC value

def var_model_selection(data):
    lag_selection_space = range(0,10)
    aic_best_lag = 0
    bic_best_lag = 0
    best_aic = 1000000
    best_bic = 1000000
    aic = []
    bic = []
    # Iterate over the predefined lag selection space
    for lag in lag_selection_space:
        print(f"Fitting VAR({lag})")
        # Fit the VAR model
        model = VAR(data)
        results = model.fit(lag)
        aic.append(results.aic)
        bic.append(results.bic)
    # If the current AIC or BIC value for the lag period is better than the previous value,
    # we can update the optimal lag period and the best AIC or BIC value.
        if results.aic < best_aic:
            best_aic = results.aic
            aic_best_lag = lag
        if results.bic < best_bic:
            best_bic = results.bic
            bic_best_lag = lag
    # Output the optimal lag period and its corresponding AIC and BIC values.
    print(f"Lag selected by AIC: {aic_best_lag}, AIC = {best_aic}")
    print(f"Lag selected by BIC: {bic_best_lag}, BIC = {best_bic}")
    # Plot the graph of AIC and BIC values as they vary with the lag period.
    plt.plot(lag_selection_space, aic, label="AIC")
    for x, y in zip(lag_selection_space, aic):
        plt.text(x, y, "{:.2f}".format(y))
    plt.plot(lag_selection_space, bic, label="BIC")
    for x, y in zip(lag_selection_space, bic):
        plt.text(x, y, "{:.2f}".format(y))
    plt.legend()
    plt.show()
```

In [13]:
```
# Apply model selection methods on the variables.
var_model_selection(var_data1)
```

```
Fitting VAR(0)
Fitting VAR(1)
Fitting VAR(2)
Fitting VAR(3)
Fitting VAR(4)
Fitting VAR(5)
Fitting VAR(6)
Fitting VAR(7)
Fitting VAR(8)
Fitting VAR(9)
Lag selected by AIC: 9, AIC = -60.01708500023309
Lag selected by BIC: 9, BIC = -42.75340220605155
```

```
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

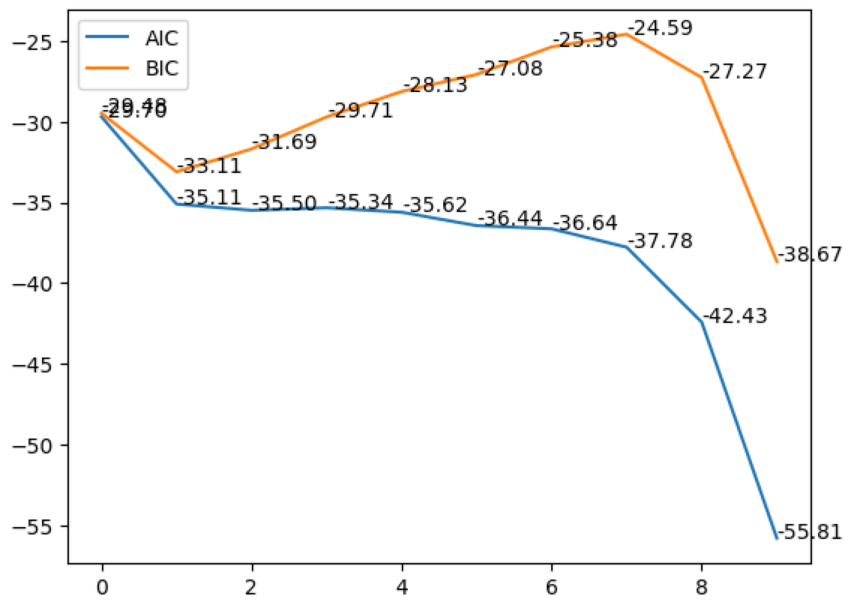


According to the AIC, we can choose local optimal p=4 for var1.

In [14]:
```
var_model_selection(var_data2)
```

```
Fitting VAR(0)
Fitting VAR(1)
Fitting VAR(2)
Fitting VAR(3)
Fitting VAR(4)
Fitting VAR(5)
Fitting VAR(6)
Fitting VAR(7)
Fitting VAR(8)
Fitting VAR(9)
Lag selected by AIC: 9, AIC = -55.81031668115683
Lag selected by BIC: 9, BIC = -38.669779603079704
```

```
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```



According to the AIC, we can choose local optimal p=1 for var2.

Fit the VAR model.

In [15]:
```
var1_lag=4
var1=VAR(var_data1)
var1_results = var1.fit(var1_lag)

#show the result
from IPython.display import display, Markdown
coefs = var1_results.params['Income inequality - Seasonal Adjusted Difference']
std_errors = var1_results.bse['Income inequality - Seasonal Adjusted Difference']
p_values = var1_results.pvalues['Income inequality - Seasonal Adjusted Difference']

results_df = pd.DataFrame({
    'Coefficient': coefs,
    'Standard Error': std_errors,
    'P-value': p_values
})
display(Markdown('### Result for Income inequality - Seasonal Adjusted Difference'))
display(results_df)
```

```
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

Result for Income inequality - Seasonal Adjusted Difference

	Coefficient	Standard Error	P-value
const	-0.001684	0.000675	0.012598
L1.Qualterly % Change of Inflation (qualterly) Difference	-0.002463	0.001883	0.190976
L1.Real GDP Change	0.000538	0.000710	0.448510
L1.Quarterly Nikkei 225 Index Change	-0.007471	0.006542	0.253473
L1.% Quarterly Change- MB (SA) Difference	0.013649	0.018421	0.458725
L1.Income inequality - Seasonal Adjusted Difference	-0.761863	0.186536	0.000044
L1.Unemployment rate - Seasonal Adjusted Difference	0.004575	0.002569	0.074960
L1.earthquake	-0.001204	0.004785	0.801281
L1.tax	0.004733	0.005187	0.361446
L2.Qualterly % Change of Inflation (qualterly) Difference	0.000907	0.001902	0.633497
L2.Real GDP Change	0.000880	0.000704	0.211027
L2.Quarterly Nikkei 225 Index Change	-0.009403	0.007443	0.206432
L2.% Quarterly Change- MB (SA) Difference	0.026545	0.019617	0.176002
L2.Income inequality - Seasonal Adjusted Difference	-0.488424	0.205977	0.017728
L2.Unemployment rate - Seasonal Adjusted Difference	0.004744	0.002578	0.065796
L2.earthquake	0.005235	0.005895	0.374521
L2.tax	0.005623	0.005133	0.273350
L3.Qualterly % Change of Inflation (qualterly) Difference	-0.000560	0.001840	0.760668
L3.Real GDP Change	0.000616	0.000667	0.355531
L3.Quarterly Nikkei 225 Index Change	-0.009629	0.007728	0.212737
L3.% Quarterly Change- MB (SA) Difference	0.023965	0.019996	0.230731
L3.Income inequality - Seasonal Adjusted Difference	-0.535670	0.200546	0.007561
L3.Unemployment rate - Seasonal Adjusted Difference	0.002866	0.002618	0.273617
L3.earthquake	-0.003015	0.006045	0.617892
L3.tax	0.007959	0.005287	0.132191
L4.Qualterly % Change of Inflation (qualterly) Difference	-0.001162	0.001646	0.480172
L4.Real GDP Change	0.001062	0.000714	0.136796
L4.Quarterly Nikkei 225 Index Change	-0.001436	0.007489	0.847975
L4.% Quarterly Change- MB (SA) Difference	0.007058	0.019935	0.723296
L4.Income inequality - Seasonal Adjusted Difference	0.223535	0.175079	0.201685
L4.Unemployment rate - Seasonal Adjusted Difference	0.003762	0.002656	0.156666
L4.earthquake	0.001271	0.004647	0.784372
L4.tax	0.004249	0.005115	0.406142

In [16]:
```
var2_lag=1
var2=VAR(var_data2)
var2_results = var2.fit(var2_lag)

#show the result
from IPython.display import display, Markdown
coefs = var2_results.params['Income inequality - Seasonal Adjusted']
std_errors = var2_results.bse['Income inequality - Seasonal Adjusted']
p_values = var2_results.pvalues['Income inequality - Seasonal Adjusted']

results_df = pd.DataFrame({
    'Coefficient': coefs,
    'Standard Error': std_errors,
    'P-value': p_values
})
display(Markdown('### Result for Income inequality - Seasonal Adjusted'))
display(results_df)
```

```
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

Result for Income inequality - Seasonal Adjusted

	Coefficient	Standard Error	P-value
const	0.120978	0.023954	4.408681e-07
L1.Qualterly % Change of Inflation (qualterly)	0.000555	0.000771	4.714966e-01
L1.Real GDP Change	0.001288	0.000706	6.818708e-02
L1.Quarterly Nikkei 225 Index Change	-0.008404	0.007550	2.656675e-01
L1.% Quarterly Change- MB (SA)	0.025452	0.017652	1.493301e-01
L1.Income inequality - Seasonal Adjusted	0.048509	0.126098	7.004625e-01
L1.Unemployment rate - Seasonal Adjusted	0.002715	0.000532	3.309594e-07
L1.earthquake	0.000301	0.004483	9.463914e-01
L1.tax	0.000400	0.006205	9.485826e-01

Part A summary, Interpret the VAR results:

从VAR_1模型来看，对于“收入不平等的一阶差分”（经过季节性调整）这一变量而言，较为显著的预测变量为前一期的“收入不平等的一阶差分”（p=0.000044）与前一期的“失业率的一阶差分”（p=0.074960）,以及前二期的“收入不平等的一阶差分”（p=0.017728）与前一期的“失业率的一阶差分”（p=0.065796）；以及前三期的“收入不平等的一阶差分”（p=0.007561)。这表明在此模型中，“收入不平等的一阶差分”的历史数据对于其当期值有显著的预测意义，同时之前的“失业率的一阶差分”也对于收入不平等具有较为显著的预测意义。然而其他变量例如通胀、GDP等宏观变量的历史信息对于“收入不平等的一阶差分”预测力度有限。

从VAR_2模型来看，对于“收入不平等”（经过季节性调整）这一变量而言，具有较大预测意义的是前一期的“实际GDP变化”（p=6.818708e-02）与前一期的“失业率”(p=3.309594e-07)，然而这有可能受到未处理线性时间趋势带来的伪回归效应影响。

总的而言，对于收入不平等的预测能够较好地依赖于其自身的历史信息以及失业率的历史信息。

Part B

In [17]:
```

from statsmodels.tsa.vector_ar.svar_model import SVAR

A1 = np.asarray([[1, 0, 0, 0, 0, 0, 0, 0],
['E', 1, 0, 0, 0, 0, 0, 0],
['E', 'E', 1, 0, 0, 0, 0, 0],
['E', 'E', 'E', 1, 0, 0, 0, 0],
['E', 'E', 'E', 'E', 1, 0, 0, 0],
['E', 'E', 'E', 'E', 'E', 1, 0, 0],
['E', 'E', 'E', 'E', 'E', 'E', 1, 0],
['E', 'E', 'E', 'E', 'E', 'E', 'E', 1]])
```

In [18]:

```
#Fit SVAR on the variables we choose
svar1 = SVAR(var_data1, svar_type="A", A=A1)
svar1_results = svar1.fit(maxlags=var1_lag)
table = pd.DataFrame(svar1_results.A,columns=["inflation_diff_col", "gdp_change_col", "index_change_col","mb_change_diff_c
                                              "income_inequal_diff_col", "unemployment_diff_col","earthquake_col", "tax_cc
                     index=["inflation_diff_col", "gdp_change_col", "index_change_col","mb_change_diff_col", "income_inequ
                            "unemployment_diff_col","earthquake_col", "tax_col"])
table
```

```
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\base\model.py:604: ConvergenceWarning: Maximum Likelihood optimization failed
to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

Out[18]:

	inflation_diff_col	gdp_change_col	index_change_col	mb_change_diff_col	income_inequal_diff_col	unemployme
inflation_diff_col	1.000000	0.000000	0.000000	0.000000	0.000000	
gdp_change_col	-0.819045	1.000000	0.000000	0.000000	0.000000	
index_change_col	0.097356	-0.027043	1.000000	0.000000	0.000000	
mb_change_diff_col	-0.007690	0.009545	-0.019697	1.000000	0.000000	
income_inequal_diff_col	0.000327	0.001419	-0.001178	-0.011064	1.000000	
unemployment_diff_col	-0.095081	0.038211	-0.059371	1.380528	-7.555124	
earthquake_col	0.022484	0.049340	-0.230622	-0.682513	-0.395556	
tax_col	-0.050972	0.021303	0.087127	-0.081797	0.127731	

In [19]:

```
#Fit SVAR on the variables without difference process
svar2 = SVAR(var_data2, svar_type="A", A=A1)
svar2_results = svar2.fit(maxlags=var2_lag)
table = pd.DataFrame(svar2_results.A,columns=["inflation_col", "gdp_change_col", "index_change_col","mb_change_col",\
                                              "income_inequal_col", "unemployment_col","earthquake_col", "tax_col"],\
                                                  index=[["inflation_col", "gdp_change_col", "index_change_col","mb_change
                                                      "income_inequal_col", "unemployment_col","earthquake_col", "tax_col
table
```

```
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\base\model.py:604: ConvergenceWarning: Maximum Likelihood optimization failed
to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

Out[19]:

	inflation_col	gdp_change_col	index_change_col	mb_change_col	income_inequal_col	unemployment_col	earthqua
inflation_col	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.(
gdp_change_col	-0.066664	1.000000	0.000000	0.000000	0.000000	0.000000	0.(
index_change_col	0.022323	-0.025683	1.000000	0.000000	0.000000	0.000000	0.(
mb_change_col	-0.002646	0.004524	-0.030328	1.000000	0.000000	0.000000	0.(
income_inequal_col	-0.000190	0.000947	-0.006281	-0.014988	1.000000	0.000000	0.(
unemployment_col	0.042638	-0.001934	-0.067673	-0.995143	-8.791983	1.000000	0.(
earthquake_col	-0.010929	0.048942	-0.089472	-1.351300	1.949420	0.009763	1.(
tax_col	-0.036270	-0.001577	0.098057	-0.383806	2.162675	-0.028555	0.(

Part B summary, Why we need SVAR on top of VAR:

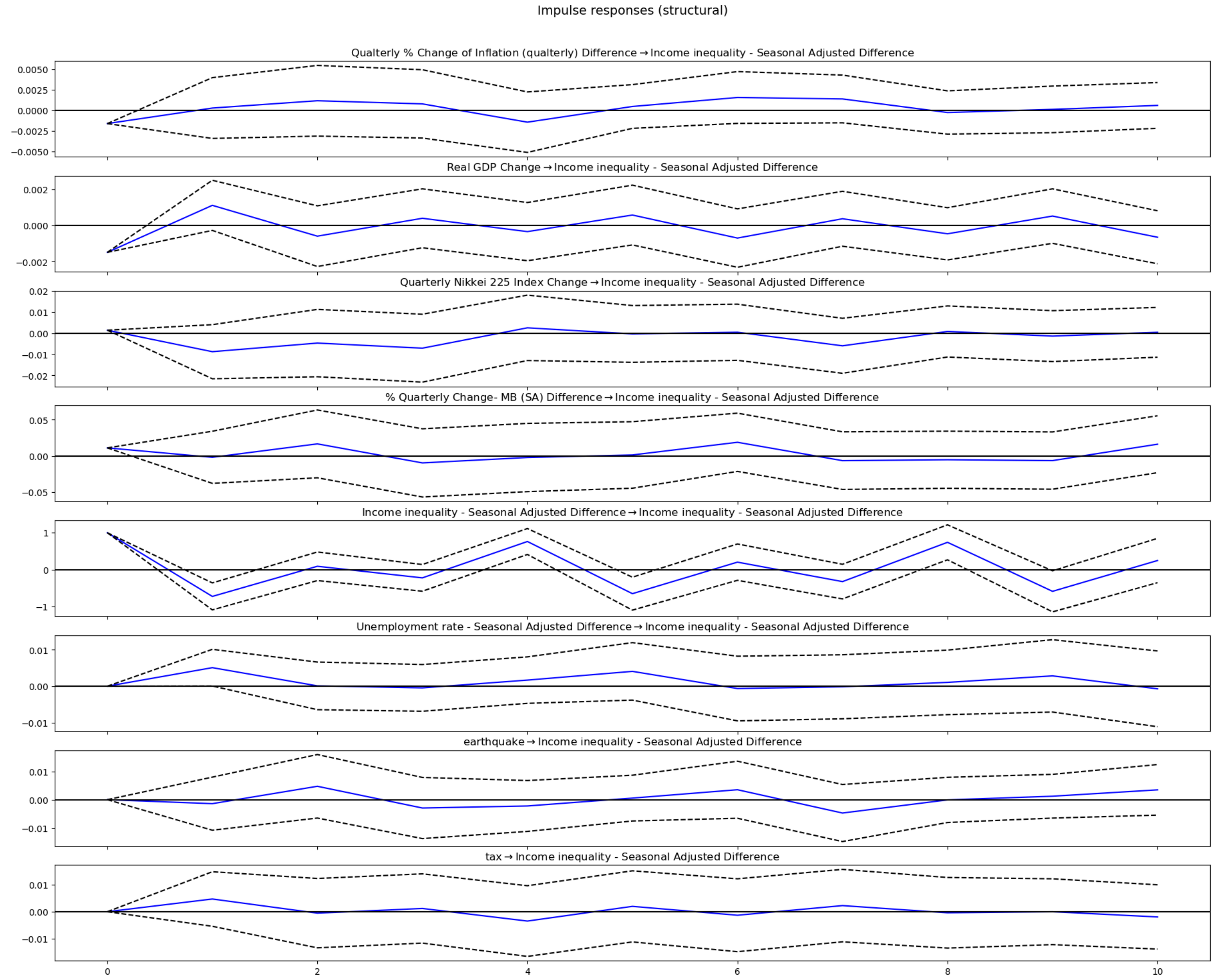
VAR模型的每个变量的误差项互相之间并不保证不相关，因此无法区分两个变量之间的关系是通过当期相互关联还是通过t对t+1,t+2...的传导的影响。SVAR模型使得变量之间的重构后误差项不再相关，可以一举两得，既通过协方差矩阵捕捉到变量之间同期的关联，也可以单纯干净地看到一个变量的外生冲击如何通过之后期进行传导对于另一些变量产生影响，也即可以作正交脉冲响应分析。

上面显示的A矩阵表示了各个宏观变量之间的当期影响系数。

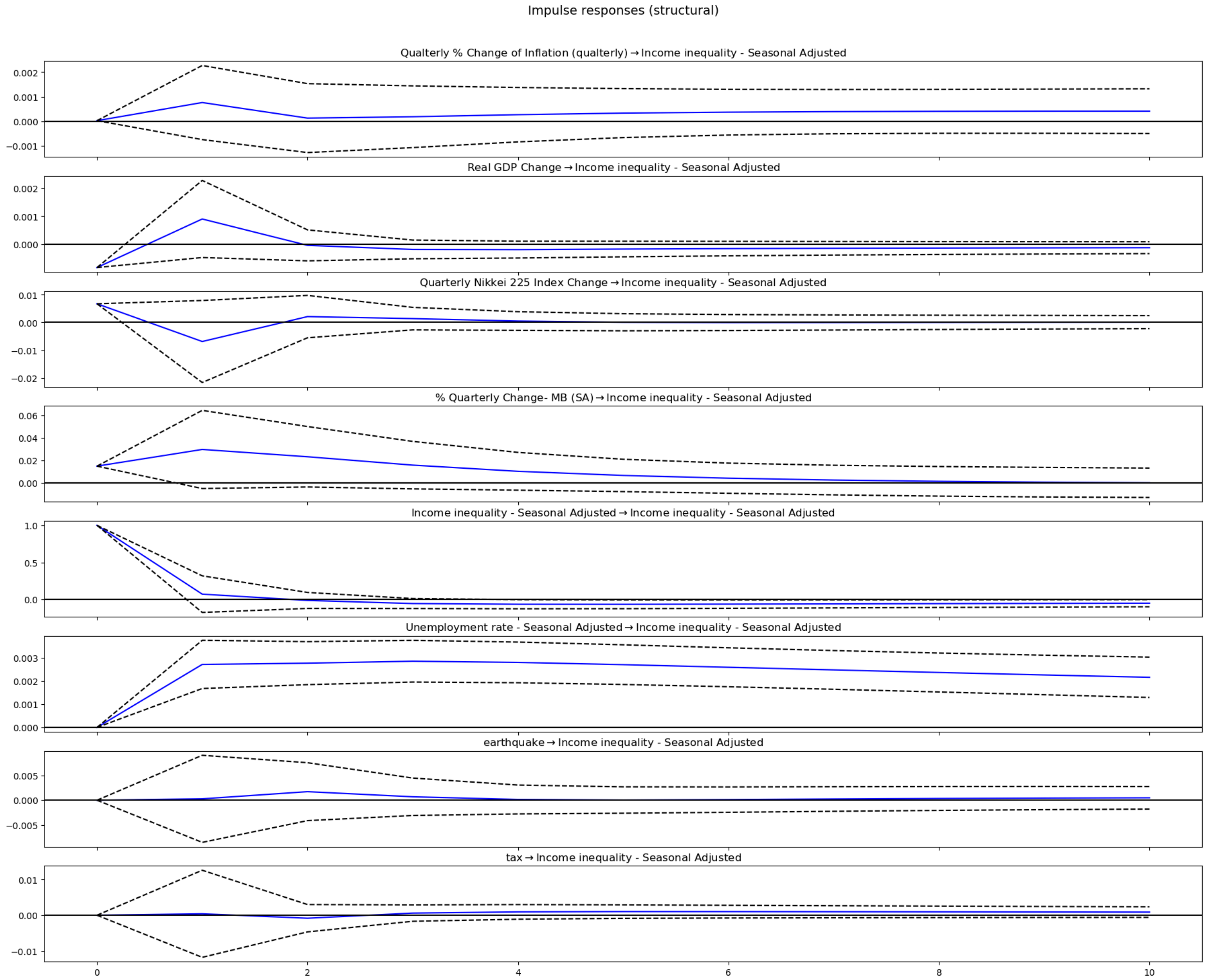
Part C

In [20]:

```
irf1 = svar1_results.irf(10)
plot1 = irf1.plot(response=income_inequal_diff_col,figsize=(20,16))
```



```
In  [21]: irf2 = svar2_results.irf(10)
          plot2 = irf2.plot(response=income_inequal_sa_col,figsize=(20,16))
```



Part C summary, impulse response plots:

上面图显示了正交脉冲响应图。

如图所示，对于"收入不平等"或者"收入不平等的一阶差分"，从模型1而言，难以看到其他变量冲击对"收入不平等一阶差分"的影响。

从模型2而言，可见"失业率"的冲击可以对"收入不平等"产生非常持续的显著的影响。同时"量宽MB"冲击可能有一期的影响（不是很显著），但是并不持续。

Part D

The ARDL model

First fit the ARDL model with series of I(1) and I(0).

In [22]:
```
from statsmodels.tsa.api import ARDL

var_data2 = df[[inflation_col,gdp_change_col,index_col,index_change_col,mb_change_col,income_inequal_sa_col,\
                unemployment_sa_col,earthquake_col,tax_col]]
ardl = ARDL(var_data2[income_inequal_sa_col], 1, var_data2[[inflation_col,gdp_change_col,index_change_col,mb_change_col,\
                                                           unemployment_sa_col,earthquake_col,tax_col]], 1, seasonal=Tru
                                                           trend="ct", period=4)
ardl_results = ardl.fit()
ardl_results.summary()
```

```
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

Out[22]:

ARDL Model Results

Dep. Variable:	Income inequality - Seasonal Adjusted	**No. Observations:**	91
Model:	Seas. ARDL(1, 1, 1, 1, 1, 1, 1, 1)	**Log Likelihood**	349.688
Method:	Conditional MLE	**S.D. of innovations**	0.005
Date:	Sat, 01 Jun 2024	**AIC**	-657.376
Time:	21:16:12	**BIC**	-604.880
Sample:	1	**HQIC**	-636.206
	91		

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0467	0.030	1.553	0.125	-0.013	0.107
trend	-0.0002	4.47e-05	-3.372	0.001	-0.000	-6.16e-05
s(2,4)	0.0003	0.002	0.155	0.877	-0.003	0.004
s(3,4)	-0.0004	0.002	-0.232	0.818	-0.004	0.003
s(4,4)	0.0001	0.002	0.072	0.943	-0.003	0.004
Income inequality - Seasonal Adjusted.L1	0.0563	0.156	0.362	0.718	-0.254	0.366
Qualterly % Change of Inflation (qualterly).L0	0.0007	0.001	0.469	0.640	-0.002	0.004
Qualterly % Change of Inflation (qualterly).L1	-0.0006	0.002	-0.422	0.674	-0.004	0.002
Real GDP Change.L0	-0.0012	0.001	-1.574	0.120	-0.003	0.000
Real GDP Change.L1	0.0011	0.001	1.531	0.130	-0.000	0.002
Quarterly Nikkei 225 Index Change.L0	0.0057	0.008	0.734	0.466	-0.010	0.021
Quarterly Nikkei 225 Index Change.L1	-0.0024	0.008	-0.287	0.775	-0.019	0.014
% Quarterly Change- MB (SA).L0	0.0344	0.024	1.435	0.156	-0.013	0.082
% Quarterly Change- MB (SA).L1	0.0241	0.024	0.994	0.323	-0.024	0.072
Unemployment rate - Seasonal Adjusted.L0	0.0019	0.003	0.737	0.464	-0.003	0.007
Unemployment rate - Seasonal Adjusted.L1	0.0027	0.003	0.891	0.376	-0.003	0.009
earthquake.L0	-0.0062	0.005	-1.142	0.258	-0.017	0.005
earthquake.L1	0.0035	0.005	0.638	0.525	-0.007	0.014
tax.L0	-0.0046	0.006	-0.750	0.456	-0.017	0.008
tax.L1	-0.0016	0.006	-0.258	0.797	-0.014	0.011

Second fit the ARDL model only with series of I(1) without I(0).

In [23]:
```
ardl = ARDL(var_data2[income_inequal_sa_col], 1, var_data2[[mb_change_col,unemployment_sa_col]], 1, seasonal=True, trend=
ardl_results = ardl.fit()
ardl_results.summary()
```

```
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

Out[23]:

ARDL Model Results

Dep. Variable:	Income inequality - Seasonal Adjusted	**No. Observations:**	91
Model:	Seas. ARDL(1, 1, 1)	**Log Likelihood**	345.212
Method:	Conditional MLE	**S.D. of innovations**	0.005
Date:	Sat, 01 Jun 2024	**AIC**	-668.424
Time:	21:16:15	**BIC**	-640.926
Sample:	1	**HQIC**	-657.335
	91		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	0.0585	0.028	2.060	0.043	0.002	0.115
trend	-0.0002	4.32e-05	-3.625	0.001	-0.000	-7.06e-05
s(2,4)	0.0007	0.002	0.420	0.675	-0.003	0.004
s(3,4)	-0.0001	0.002	-0.082	0.935	-0.003	0.003
s(4,4)	0.0006	0.002	0.341	0.734	-0.003	0.004
Income inequality - Seasonal Adjusted.L1	-0.0290	0.145	-0.200	0.842	-0.317	0.259
% Quarterly Change- MB (SA).L0	0.0328	0.022	1.492	0.140	-0.011	0.077
% Quarterly Change- MB (SA).L1	0.0239	0.022	1.093	0.278	-0.020	0.067
Unemployment rate - Seasonal Adjusted.L0	0.0012	0.002	0.487	0.627	-0.004	0.006
Unemployment rate - Seasonal Adjusted.L1	0.0036	0.003	1.262	0.211	-0.002	0.009

Part D summary, fit the ARDL:

上面的表结果显示了使用I(1)和I(0)序列以及只用I(1)序列进行的ARDL模型结果。

然而根据模型的估计,我并没有得到很显著的系数.

Part E

First we use Engle Granger Test

In [236]:
```
from arch.unitroot import engle_granger
test1 = engle_granger(var_data2[income_inequal_sa_col], var_data2[[mb_change_col,unemployment_sa_col]], trend="ct")
test1
```

Out[236]:

Engle-Granger Cointegration Test

Test Statistic	-2.183
P-value	0.827
ADF Lag length	3
Estimated Root ρ (γ+1)	0.660

Trend: Constant
Critical Values: -3.96 (10%), -4.28 (5%), -4.91 (1%)
Null Hypothesis: No Cointegration
Alternative Hypothesis: Cointegration
Distribution Order: 2

Second we use ARDL Test

In [24]:
```python
from statsmodels.tsa.ardl import BoundsTestResult
from statsmodels.tsa.ardl import UECM
```

use I(0) and I(1)

In [25]:
```python
var_data2 = df[[inflation_col,gdp_change_col,index_col,index_change_col,mb_change_col,income_inequal_sa_col,\
                unemployment_sa_col,earthquake_col,tax_col]]
uecm = UECM(var_data2[income_inequal_sa_col], 1, var_data2[[inflation_col,gdp_change_col,index_change_col,mb_change_col,\
                                                            unemployment_sa_col,earthquake_col,tax_col]] , 1, trend='ct',
                                                            period=4)

uecm_results = uecm.fit()
uecm_results.summary()
boundsTestResult = uecm_results.bounds_test(case=1)
boundsTestResult
```

```
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

Out[25]:
```
BoundsTestResult
Stat: 2.33805
Upper P-value: 0.232
Lower P-value: 0.014
Null: No Cointegration
Alternative: Possible Cointegration
```

use only I(1)

In [26]:
```python
var_data2 = df[[inflation_col,gdp_change_col,index_col,index_change_col,mb_change_col,income_inequal_sa_col,\
                unemployment_sa_col,earthquake_col,tax_col]]
uecm = UECM(var_data2[income_inequal_sa_col], 1, var_data2[[index_col,mb_change_col,unemployment_sa_col]]\
            , 1, trend='ct', seasonal=True,  period=4)
uecm_results = uecm.fit()
uecm_results.summary()
boundsTestResult = uecm_results.bounds_test(case=1)
boundsTestResult
```

```
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

Out[26]:
```
BoundsTestResult
Stat: 5.37683
Upper P-value: 0.00186
Lower P-value: 8.52e-05
Null: No Cointegration
Alternative: Possible Cointegration
```

Part E summary, test the ways of cointegration among those I(1) sequences:

上面的结果显示了I(1)序列的Engle Granger Test与ARDL Test(bound test)

前者的P值为0.827,显示不存在协整关系.

然而ARDL Test(bound test)的Upper bound的p值为0.00186,显示可能存在协整关系.

Part F

In [27]:

```
from statsmodels.tsa.vector_ar import vecm
from statsmodels.tsa.vector_ar.vecm import coint_johansen
# Select the order of the VECM
vecm_lag = vecm.select_order(data=var_data2[[income_inequal_sa_col,mb_change_col,unemployment_sa_col]],maxlags=10)
# Display the result
vecm_lag.summary()
```

```
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

Out[27]:

VECM Order Selection (* highlights the minimums)

	AIC	BIC	FPE	HQIC
0	-20.15	-19.79*	1.774e-09	-20.01
1	-20.28	-19.66	1.554e-09	-20.03
2	-20.24	-19.35	1.625e-09	-19.88
3	-20.43	-19.26	1.358e-09	-19.96
4	-21.04*	-19.61	7.409e-10*	-20.47*
5	-20.88	-19.18	8.805e-10	-20.20
6	-20.88	-18.91	8.970e-10	-20.09
7	-20.96	-18.72	8.456e-10	-20.06
8	-20.86	-18.36	9.535e-10	-19.86
9	-20.78	-18.01	1.076e-09	-19.67
10	-20.65	-17.61	1.281e-09	-19.43

In [28]:
```
# Perform the Johansen cointegration rank test
from statsmodels.tsa.vector_ar.vecm import coint_johansen
coint_result = coint_johansen(var_data2[[income_inequal_sa_col,mb_change_col,unemployment_sa_col]], 1, 4)
#Trace Statistics
print("Trace Statistics:")
print(coint_result.lr1)
#Maximum Eigenvalue Statistic
print("Maximum Eigenvalue Statistic:")
print(coint_result.lr2)
# From the statistics compared to the critical value(not shown), we believe that there are 2 cointegration relationships e
print("Trace Statistics:", coint_result.lr1)
print("Critical Values (90%, 95%, 99%):", coint_result.cvt)
print("Eigenvalues:", coint_result.eig)
print("Maximum Eigenvalue Statistics:", coint_result.lr2)
print("Critical Values for max test (90%, 95%, 99%):", coint_result.cvm)
```

```
Trace Statistics:
[38.63210302 17.17009736  3.35936925]
Maximum Eigenvalue Statistic:
[21.46200565 13.81072811  3.35936925]
Trace Statistics: [38.63210302 17.17009736  3.35936925]
Critical Values (90%, 95%, 99%): [[32.0645 35.0116 41.0815]
 [16.1619 18.3985 23.1485]
 [ 2.7055  3.8415  6.6349]]
Eigenvalues: [0.22085507 0.14835871 0.03830933]
Maximum Eigenvalue Statistics: [21.46200565 13.81072811  3.35936925]
Critical Values for max test (90%, 95%, 99%): [[21.8731 24.2522 29.2631]
 [15.0006 17.1481 21.7465]
 [ 2.7055  3.8415  6.6349]]
```

We use r=2 according to Trace Statistics at Critical Values of 90%, and we can also try r=1 according to Trace Statistics at Critical Values of 95%.

In [29]:

```
#Fit vecm model use lag=1 and rank=2
vecm_lag = 4
coint_rank = 2
vecm_model = vecm.VECM(var_data2[[income_inequal_sa_col,mb_change_col,unemployment_sa_col]], k_ar_diff=vecm_lag, coint_ra
vecm_results = vecm_model.fit()
vecm_results.summary()
```

```
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

Out[29]:

Det. terms outside the coint. relation & lagged endog. parameters for equation Income inequality - Seasonal Adjusted

	coef	std err	z	P>|z|	[0.025	0.975]
L1.Income inequality - Seasonal Adjusted	-0.5584	0.143	-3.914	0.000	-0.838	-0.279
L1.% Quarterly Change- MB (SA)	0.0008	0.017	0.048	0.961	-0.032	0.034
L1.Unemployment rate - Seasonal Adjusted	0.0043	0.002	2.327	0.020	0.001	0.008
L2.Income inequality - Seasonal Adjusted	-0.3149	0.153	-2.055	0.040	-0.615	-0.015
L2.% Quarterly Change- MB (SA)	0.0103	0.017	0.622	0.534	-0.022	0.043
L2.Unemployment rate - Seasonal Adjusted	0.0037	0.002	1.886	0.059	-0.000	0.008
L3.Income inequality - Seasonal Adjusted	-0.4575	0.151	-3.038	0.002	-0.753	-0.162
L3.% Quarterly Change- MB (SA)	0.0019	0.016	0.119	0.905	-0.029	0.032
L3.Unemployment rate - Seasonal Adjusted	0.0020	0.002	1.018	0.309	-0.002	0.006
L4.Income inequality - Seasonal Adjusted	0.2292	0.139	1.651	0.099	-0.043	0.501
L4.% Quarterly Change- MB (SA)	-0.0125	0.015	-0.844	0.399	-0.041	0.017
L4.Unemployment rate - Seasonal Adjusted	0.0031	0.002	1.594	0.111	-0.001	0.007

Det. terms outside the coint. relation & lagged endog. parameters for equation % Quarterly Change- MB (SA)

	coef	std err	z	P>|z|	[0.025	0.975]
L1.Income inequality - Seasonal Adjusted	2.7415	0.986	2.781	0.005	0.809	4.674
L1.% Quarterly Change- MB (SA)	-0.1270	0.116	-1.091	0.275	-0.355	0.101
L1.Unemployment rate - Seasonal Adjusted	-0.0317	0.013	-2.453	0.014	-0.057	-0.006
L2.Income inequality - Seasonal Adjusted	2.5149	1.059	2.375	0.018	0.440	4.590
L2.% Quarterly Change- MB (SA)	-0.1939	0.115	-1.689	0.091	-0.419	0.031
L2.Unemployment rate - Seasonal Adjusted	-0.0164	0.013	-1.220	0.222	-0.043	0.010
L3.Income inequality - Seasonal Adjusted	2.1238	1.040	2.041	0.041	0.085	4.163
L3.% Quarterly Change- MB (SA)	-0.1412	0.107	-1.318	0.188	-0.351	0.069
L3.Unemployment rate - Seasonal Adjusted	-0.0090	0.014	-0.652	0.515	-0.036	0.018
L4.Income inequality - Seasonal Adjusted	2.6028	0.959	2.714	0.007	0.723	4.483
L4.% Quarterly Change- MB (SA)	-0.1501	0.102	-1.469	0.142	-0.350	0.050
L4.Unemployment rate - Seasonal Adjusted	0.0089	0.014	0.655	0.512	-0.018	0.036

Det. terms outside the coint. relation & lagged endog. parameters for equation Unemployment rate - Seasonal Adjusted

	coef	std err	z	P>|z|	[0.025	0.975]
L1.Income inequality - Seasonal Adjusted	-4.2978	6.944	-0.619	0.536	-17.908	9.313
L1.% Quarterly Change- MB (SA)	2.0177	0.820	2.460	0.014	0.410	3.625
L1.Unemployment rate - Seasonal Adjusted	-0.0124	0.091	-0.136	0.892	-0.191	0.166
L2.Income inequality - Seasonal Adjusted	3.2212	7.457	0.432	0.666	-11.394	17.836
L2.% Quarterly Change- MB (SA)	0.9834	0.809	1.216	0.224	-0.601	2.568
L2.Unemployment rate - Seasonal Adjusted	-0.0261	0.095	-0.275	0.783	-0.212	0.160
L3.Income inequality - Seasonal Adjusted	8.5542	7.328	1.167	0.243	-5.809	22.918
L3.% Quarterly Change- MB (SA)	1.4972	0.755	1.984	0.047	0.018	2.976
L3.Unemployment rate - Seasonal Adjusted	0.0427	0.097	0.440	0.660	-0.147	0.233
L4.Income inequality - Seasonal Adjusted	12.3515	6.755	1.828	0.067	-0.889	25.592
L4.% Quarterly Change- MB (SA)	0.7132	0.720	0.991	0.322	-0.698	2.124
L4.Unemployment rate - Seasonal Adjusted	0.5117	0.096	5.345	0.000	0.324	0.699

Loading coefficients (alpha) for equation Income inequality
- Seasonal Adjusted

	coef	std err	z	P>|z|	[0.025	0.975]
ec1	-0.1953	0.065	-3.007	0.003	-0.323	-0.068
ec2	0.0082	0.014	0.597	0.550	-0.019	0.035

Loading coefficients (alpha) for equation % Quarterly Change- MB (SA)

	coef	std err	z	P>|z|	[0.025	0.975]
ec1	0.0922	0.449	0.205	0.837	-0.788	0.972
ec2	-0.2672	0.095	-2.799	0.005	-0.454	-0.080

Loading coefficients (alpha) for equation Unemployment rate - Seasonal Adjusted

	coef	std err	z	P>|z|	[0.025	0.975]
ec1	2.0951	3.161	0.663	0.508	-4.101	8.291
ec2	-0.9101	0.672	-1.353	0.176	-2.228	0.408

Cointegration relations for loading-coefficients-column 1

	coef	std err	z	P>|z|	[0.025	0.975]
beta.1	1.0000	0	0	0.000	1.000	1.000
beta.2	0	0	0	0.000	0	0
beta.3	-0.0058	5.31e-05	-109.993	0.000	-0.006	-0.006

Cointegration relations for loading-coefficients-column 2

	coef	std err	z	P>|z|	[0.025	0.975]
beta.1	0	0	0	0.000	0	0
beta.2	1.0000	0	0	0.000	1.000	1.000
beta.3	-0.0007	0.000	-2.975	0.003	-0.001	-0.000

In [30]:

```
#Fit vecm model use lag=1 and rank=1
vecm_lag = 4
coint_rank = 1
vecm_model = vecm.VECM(var_data2[[income_inequal_sa_col,mb_change_col,unemployment_sa_col]], k_ar_diff=vecm_lag, coint_ra
vecm_results = vecm_model.fit()
vecm_results.summary()
```

```
D:\anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided an
d will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

Out[30]:

Det. terms outside the coint. relation & lagged endog. parameters for equation Income inequality - Seasonal Adjusted

	coef	std err	z	P>|z|	[0.025	0.975]
L1.Income inequality - Seasonal Adjusted	-0.6118	0.143	-4.281	0.000	-0.892	-0.332
L1.% Quarterly Change- MB (SA)	-0.0107	0.016	-0.665	0.506	-0.042	0.021
L1.Unemployment rate - Seasonal Adjusted	0.0047	0.002	2.472	0.013	0.001	0.008
L2.Income inequality - Seasonal Adjusted	-0.3405	0.156	-2.185	0.029	-0.646	-0.035
L2.% Quarterly Change- MB (SA)	0.0008	0.016	0.049	0.961	-0.031	0.033
L2.Unemployment rate - Seasonal Adjusted	0.0038	0.002	1.895	0.058	-0.000	0.008
L3.Income inequality - Seasonal Adjusted	-0.4737	0.154	-3.086	0.002	-0.775	-0.173
L3.% Quarterly Change- MB (SA)	-0.0049	0.015	-0.318	0.751	-0.035	0.025
L3.Unemployment rate - Seasonal Adjusted	0.0020	0.002	1.000	0.318	-0.002	0.006
L4.Income inequality - Seasonal Adjusted	0.2322	0.142	1.639	0.101	-0.046	0.510
L4.% Quarterly Change- MB (SA)	-0.0177	0.015	-1.189	0.235	-0.047	0.011
L4.Unemployment rate - Seasonal Adjusted	0.0032	0.002	1.580	0.114	-0.001	0.007

Det. terms outside the coint. relation & lagged endog. parameters for equation % Quarterly Change-MB (SA)

	coef	std err	z	P>|z|	[0.025	0.975]
L1.Income inequality - Seasonal Adjusted	2.3771	0.987	2.408	0.016	0.443	4.312
L1.% Quarterly Change- MB (SA)	-0.2056	0.111	-1.852	0.064	-0.423	0.012
L1.Unemployment rate - Seasonal Adjusted	-0.0293	0.013	-2.235	0.025	-0.055	-0.004
L2.Income inequality - Seasonal Adjusted	2.3399	1.076	2.174	0.030	0.230	4.450
L2.% Quarterly Change- MB (SA)	-0.2590	0.112	-2.317	0.021	-0.478	-0.040
L2.Unemployment rate - Seasonal Adjusted	-0.0158	0.014	-1.148	0.251	-0.043	0.011
L3.Income inequality - Seasonal Adjusted	2.0127	1.060	1.898	0.058	-0.065	4.091
L3.% Quarterly Change- MB (SA)	-0.1873	0.106	-1.759	0.079	-0.396	0.021
L3.Unemployment rate - Seasonal Adjusted	-0.0089	0.014	-0.636	0.525	-0.036	0.019
L4.Income inequality - Seasonal Adjusted	2.6235	0.979	2.680	0.007	0.705	4.542
L4.% Quarterly Change- MB (SA)	-0.1854	0.103	-1.808	0.071	-0.386	0.016
L4.Unemployment rate - Seasonal Adjusted	0.0092	0.014	0.661	0.509	-0.018	0.036

Det. terms outside the coint. relation & lagged endog. parameters for equation Unemployment rate - Seasonal Adjusted

	coef	std err	z	P>|z|	[0.025	0.975]
L1.Income inequality - Seasonal Adjusted	-4.9412	6.820	-0.725	0.469	-18.308	8.425
L1.% Quarterly Change- MB (SA)	1.8790	0.767	2.448	0.014	0.375	3.383
L1.Unemployment rate - Seasonal Adjusted	-0.0082	0.091	-0.091	0.928	-0.186	0.169
L2.Income inequality - Seasonal Adjusted	2.9122	7.438	0.392	0.695	-11.666	17.490
L2.% Quarterly Change- MB (SA)	0.8684	0.773	1.124	0.261	-0.646	2.382
L2.Unemployment rate - Seasonal Adjusted	-0.0250	0.095	-0.263	0.793	-0.211	0.161
L3.Income inequality - Seasonal Adjusted	8.3581	7.326	1.141	0.254	-6.001	22.717
L3.% Quarterly Change- MB (SA)	1.4158	0.736	1.924	0.054	-0.026	2.858
L3.Unemployment rate - Seasonal Adjusted	0.0427	0.097	0.440	0.660	-0.148	0.233
L4.Income inequality - Seasonal Adjusted	12.3880	6.764	1.831	0.067	-0.869	25.645
L4.% Quarterly Change- MB (SA)	0.6509	0.709	0.918	0.358	-0.738	2.040
L4.Unemployment rate - Seasonal Adjusted	0.5122	0.096	5.343	0.000	0.324	0.700

Loading coefficients (alpha) for equation Income inequality - Seasonal Adjusted

	coef	std err	z	P>|z|	[0.025	0.975]
ec1	-0.1099	0.048	-2.277	0.023	-0.205	-0.015

Loading coefficients (alpha) for equation % Quarterly

Change- MB (SA)

	coef	std err	z	P>|z|	[0.025	0.975]
ec1	0.6753	0.334	2.025	0.043	0.022	1.329

Loading coefficients (alpha) for equation Unemployment rate - Seasonal Adjusted

	coef	std err	z	P>|z|	[0.025	0.975]
ec1	3.1246	2.305	1.356	0.175	-1.392	7.641

Cointegration relations for loading-coefficients-column 1

	coef	std err	z	P>|z|	[0.025	0.975]
beta.1	1.0000	0	0	0.000	1.000	1.000
beta.2	-0.2268	0.073	-3.107	0.002	-0.370	-0.084
beta.3	-0.0057	7.54e-05	-75.303	0.000	-0.006	-0.006

Part F summary:

上面的Trace Statistics显示变量之间存在1种(95%显著性水平)或者2种(90%显著性水平)协整关系。

同时上面的表结果Cointegration relations for loading-coefficients展示了协整系数,即表明三个变量之间存在着长期关系,

rank=2的beta系数分别为(1.0000 -3.298e-17 -0.0058)和(2.316e-16 1.0000 -0.0007) rank=1的beta系数分别为(1.0000 -0.2268 -0.0057)

且这些协整系数都显著.

以rank=1为例子,三个变量的长期均衡关系可以表示为income_inequal=(-0.2268)mb_change+(-0.0057)unemployment_rate+常数项/误差项

END