例 5.12 Python 程序代码

```
# 模型相关包
import statsmodels.api as sm
import statsmodels.stats.diagnostic
# 画图包
import matplotlib.pyplot as plt
# 其他包
import pandas as pd
import numpy as np
plt.plot(changeXAUUSD,'r',label='XAU USD')
plt.plot(shfeXAU,'g',label='SHFE XAU')
plt.title('Correlation: ' + str(correlation))
plt.grid(True)
plt.axis('tight')
plt.legend(loc=0)
plt.ylabel('Price')
plt.show()
adfResult = sm.tsa.stattools.adfuller(data,maxlags)
output = pd.DataFrame(index=['Test Statistic Value', "p-value", "Lags Used", "Number
of Observations Used",
                                        "Critical     Value(1%)",     "Critical
Value(5%)", "Critical Value(10%)"],
                                    columns=['value'])
 output['value']['Test Statistic Value'] = adfResult[0]
 output['value']['p-value'] = adfResult[1]
 output['value']['Lags Used'] = adfResult[2]
 output['value']['Number of Observations Used'] = adfResult[3]
 output['value']['Critical Value(1%)'] = adfResult[4]['1%']
 output['value']['Critical Value(5%)'] = adfResult[4]['5%']
 output['value']['Critical Value(10%)'] = adfResult[4]['10%']
result = sm.tsa.stattools.coint(data1,data2)

lnDataDict = {'lnSHFEDiff':lnSHFEDiff,'lnXAUDiff':lnXAUDiff}
lnDataDictSeries = pd.DataFrame(lnDataDict,index=lnSHFEDiffIndex)
data = lnDataDictSeries[['lnSHFEDiff','lnXAUDiff']]
#建立对象，dataframe 就是前面的 data，varLagNum 就是你自己定的滞后阶数
orgMod = sm.tsa.VARMAX(dataframe,order=(varLagNum,0),trend='nc',exog=None)
#估计：就是模型
fitMod = orgMod.fit(maxiter=1000,disp=False)
# 打印统计结果
print(fitMod.summary())
# 获得模型残差
resid = fitMod.resid
result = {'fitMod':fitMod,'resid':resid}
```

```
# 原假设：无漂移（平稳），备择假设：有漂移（不平稳）
result = statsmodels.stats.diagnostic.breaks_cusumolsresid(resid)

# orthogonalized=True，代表采用乔里斯基正交
ax = fitMod.impulse_responses(terms, orthogonalized=True).plot(figsize=(12, 8))
plt.show()

md = sm.tsa.VAR(dataFrame)
re = md.fit(2)
fevd = re.fevd(10)
# 打印出方差分解的结果
print(fevd.summary())
# 画图
fevd.plot(figsize=(12, 16))
plt.show()
```