

FBDQA-课程编号-01510313

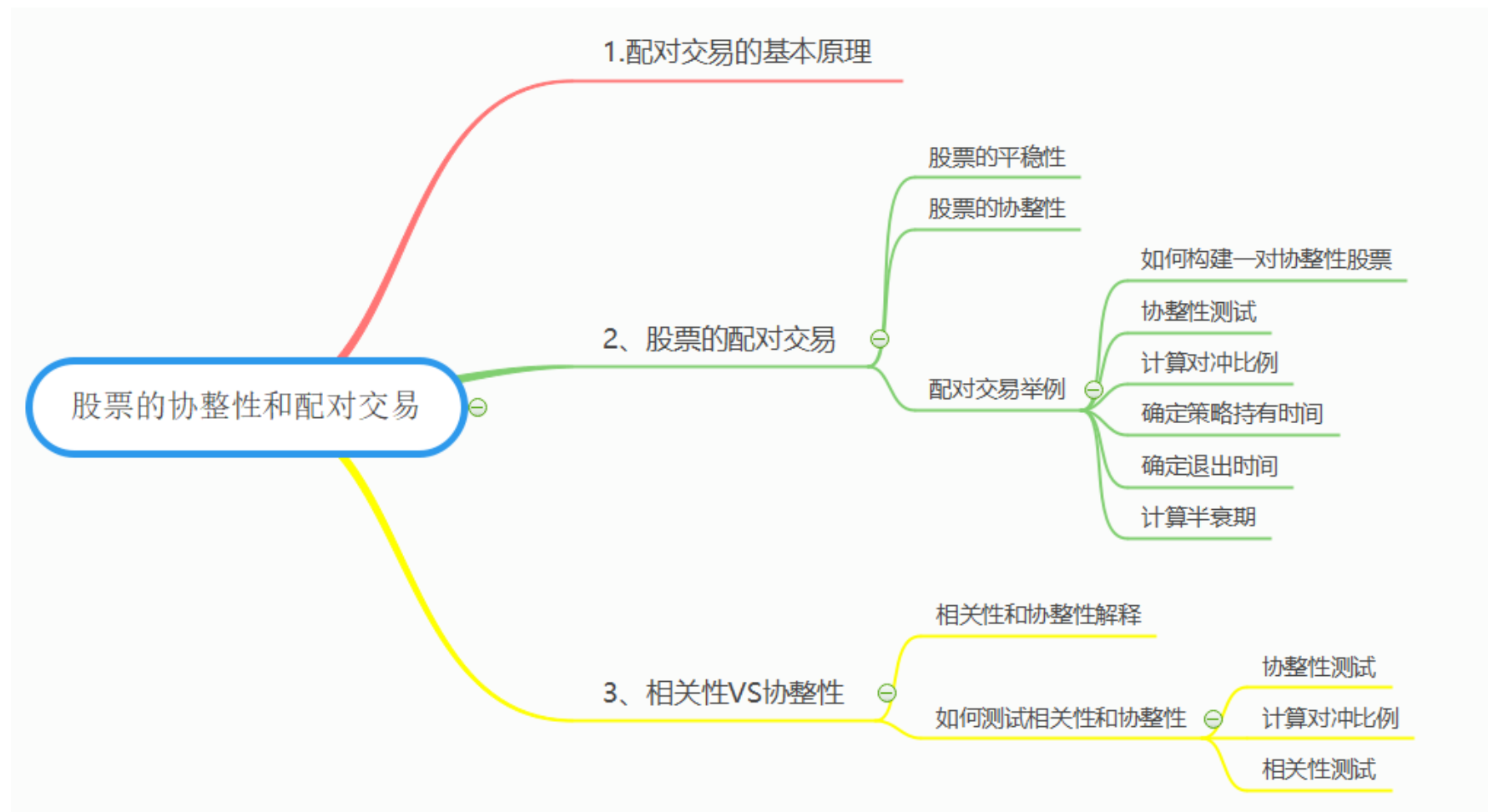


# 配对交易和协整性

智能系统实验室

清华大学iCenter

# 导学



# 名词术语

- 平稳性 (stationarity)
- 协整性 (cointegration)
- 对冲比例 (hedge ratio)
- 多空价差 (spread)
- 策略持有时间 (holding time)
- 相关性 (correlation)
- 收益率 (return)

# 目录

## 一、配对交易

- 为什么？
- 怎么做？
- GLD与GDX实例

## 二、相关性 VS. 协整性

- K0与PEP实例

参考书：[1] Ernest P. Chan, Quantitative Trading, Wiley 2008.  
notebook:example7\_\*.ipynb

# 配对交易（Pairs Trading）的基本原理

- 在量化交易部分的第八讲，介绍了配对交易进行套利
- 国外配对交易已经广泛应用股票交易中，在融资融券发展到一定程度后，相信在A股市场中也会得到广泛应用。
- 配对交易的基础是市场交易品种之间的协整性，以股票型配对交易为例。
- 相关概念：平稳性、协整性、对冲比例、Spread、策略持有时间、相关性。

# 为什么做股票的配对交易？

- 原因：单只股票不具有平稳性，利用同一行业的两只股票之间的协整性可以构建一个均值回复的策略。
- 配对交易的要素：
  - 对冲比例 (hedge ratio)
  - 价格差 (Spread)
  - 策略持有时间 (half-life)

# 如何做配对交易策略？

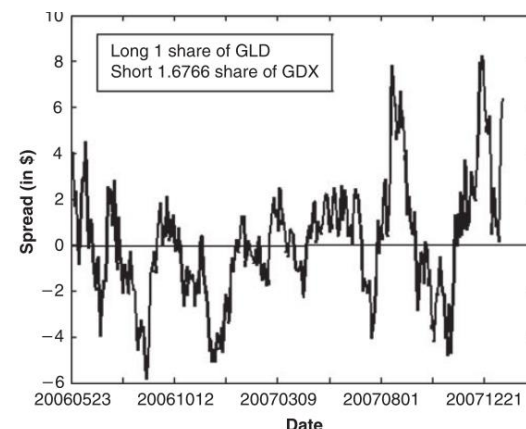
- 步骤一：先进行测试，找到可以配对的交易对象-协整性测试
- 步骤二：持有比例---对冲比例
- 步骤三：持有的时间---计算半衰期
- 步骤四：策略退出

# 一、配对交易



# 股票的平稳性

- 如果时间序列从未偏离其初始值，那么它就是“平稳的”。
- 很明显，如果一种股票的价格序列是平稳的，那么它将是均值回归策略的一个很好的选择。
- 但可惜的是，大多数股票价格序列都不是平稳的——它们表现出几何上的随机游走，使得它们离起始点的价格越来越远。

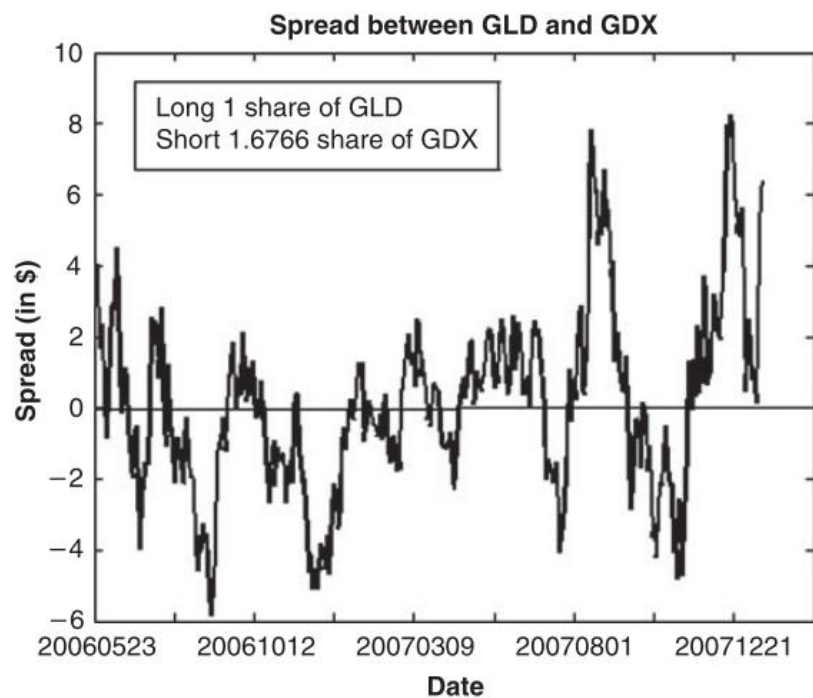


# 股票的协整性

- 可以找到一对股票，如果做多其中一只股票而做空另一只股票，整个投资组合的市场价值是稳定的。
- 在这种情况下，这两个单独的时间序列就是协整的。通常，形成协整对的两只股票来自**同一个行业**。
- 交易员们很早就熟悉这种配对交易策略。他们在这些对形成的股票价格差价较低时买入对组合，在差价较高时卖出/做空对组合——换句话说，这是一种经典的均值回归策略。

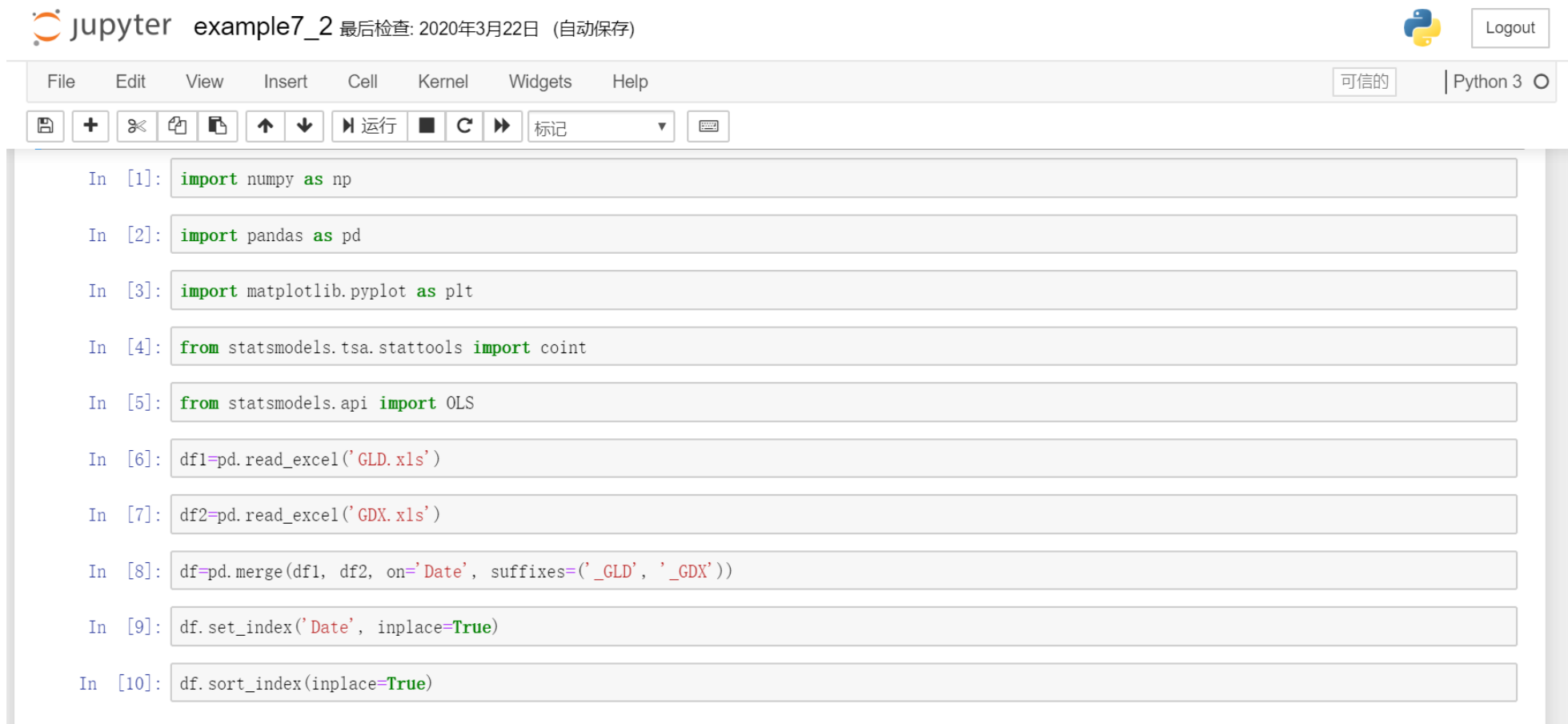
# 配对交易案例

- GLD (Gold ETF) versus GDX (Gold miners).



**FIGURE 7.4** A Stationary Time Series Formed by the Spread between GLD and GDX

# 步骤一：构建具有协整性的一对股票



The image shows a Jupyter Notebook interface with the title "example7\_2" and a timestamp "最后检查: 2020年3月22日 (自动保存)". The interface includes a top menu bar with options like File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu is a toolbar with icons for saving, adding cells, and running code. The main area contains 10 code cells, each starting with "In [n]:" followed by Python code. The code imports numpy, pandas, and matplotlib, then reads two Excel files ('GLD.xls' and 'GDX.xls'), merges them on the 'Date' column, and sorts the resulting DataFrame by index.

```
In [1]: import numpy as np

In [2]: import pandas as pd

In [3]: import matplotlib.pyplot as plt

In [4]: from statsmodels.tsa.stattools import coint

In [5]: from statsmodels.api import OLS

In [6]: df1=pd.read_excel('GLD.xls')

In [7]: df2=pd.read_excel('GDX.xls')

In [8]: df=pd.merge(df1, df2, on='Date', suffixes=('_GLD', '_GDX'))

In [9]: df.set_index('Date', inplace=True)

In [10]: df.sort_index(inplace=True)
```

(example7\_2.ipynb)

# 步骤一：协整性测试

## 恩格尔-格兰杰测试

### Run cointegration (Engle-Granger) test

```
In [11]: coint_t, pvalue, crit_value=coint(df['Adj Close_GLD'], df['Adj Close_GDX'])
```

```
In [12]: (coint_t, pvalue, crit_value) # abs(t-stat) > critical value at 95%. pvalue says probability of null hypothesis (of no cointegration) is only
```

```
Out[12]: (-3.698116076330047,  
          0.018427835409538105,  
          array([-3.92518794, -3.35208799, -3.05551324]))
```

$|coint\_t| > 1.96$ ,  $pvalue < 0.05$

结论：在95%的置信概率下具有协整性

## 步骤二： 计算对冲比例

### Determine hedge ratio

```
In [13]: model=OLS(df['Adj Close_GLD'], df['Adj Close_GDX'])
```

```
In [14]: results=model.fit()
```

```
In [15]: hedgeRatio=results.params
```

```
In [16]: hedgeRatio
```

```
Out[16]: Adj Close_GDX    1.639523  
dtype: float64
```

# 价格差 (Spread) 序列示意图



$$\text{spread} = \text{GLD} - \text{hedgeRatio} * \text{GDX}$$

价格差 (Spread) 的含义是什么?

- ☒ A 做多一只股票和做空另一只股票的价差
- ☐ B 单一股票的买卖价差

提交



## 步骤三&四： 确定策略持有时间

- 最佳持有时间通常会随着信息传播速度的加快和抓住这个交易机会的交易员数量的增加而减少。
- 有效市场假说：信息的传播
- 套利机会的消失：不再存在错误定价
- 策略有效性的变化、市场环境的变化
- 以下有一种比较可靠的统计方法来确定最佳持有期，而不依赖于有限的实际交易数量。

为什么要确定策略持有时间？

- ☒ A 有效市场假说：信息的传播
- ☒ B 套利机会的消失，即不再存在错误定价
- ☒ C 策略有效性的变化
- ☒ D 市场环境本身的变化

提交

# 计算策略持有时间

- Ornstein-Uhlenbeck公式:

$$dz(t) = -\theta(z(t) - \mu)dt + dW$$

- $z(t)$ 是投资组合的价格时间序列，也就是价格差序列（spread）。
  - $\mu$ 是 $z(t)$ 的历史均值， $dW$ 是一个随机的高斯噪声。
- 给定 $z(t)$ 的时间序列值，我们可以很容易地找到 $\theta(\mu)$ 。
- $z(t)$ 随着时间衰减，其半衰期（halflife）等于 $\ln(2) / \theta$ ，这是指预期的时间扩散到初始平均偏差的一半。
- 此半衰期可用于确定均值回复位置的最佳保持时间。

# 计算持有时间

- 半衰期更稳健的估计可以直接从交易模型得到。
- 可以利用整个时间序列，找到最好的 $\theta$ 的估计，半衰期 (half-life) 等于  $\ln(2) / \theta$
- 在本例子中，我们使用GLD和GDX之间的差值来估计均值回归的半衰期。

# 计算均值回复时间序列的半衰期 (example7\_5.ipynb)

```
In [47]: prevz=z.shift()
```

```
In [48]: dz=z-prevz
```

```
In [49]: dz=dz[1:,]
```

```
In [50]: prevz=prevz[1:,]
```

```
In [51]: model2=OLS(dz, prevz-np.mean(prevz))
```

```
In [52]: results2=model2.fit()
```

```
In [53]: theta=results2.params
```

```
In [54]: theta
```

```
Out[54]: x1    -0.088423  
         dtype: float64
```

```
In [55]: halflife=-np.log(2)/theta
```

```
In [56]: halflife
```

```
Out[56]: x1      7.839031  
         dtype: float64
```

## 二、相关性 VS. 协整性

# 相关性 VS. 协整性

- 许多交易者不熟悉平稳性和协整性的概念。
- 但是他们中的大多数人都熟悉相关性。
- 从表面上看，相关性似乎等同于协整性。事实上，它们很不一样。

# 相关性

- 两个价格序列之间的相关性实际上是指它们在一段时间内(具体来说, 假设是一天)收益率的相关性。
- 如果两只股票是正相关的, 那么它们的价格很有可能在大多数日子里朝着同一个方向变动。
- 然而, 正相关并不能说明这两只股票的长期表现。特别是, 这并不能保证股票价格在长期内不会变得越来越差, 即使它们在大多数日子里朝着同一个方向移动。

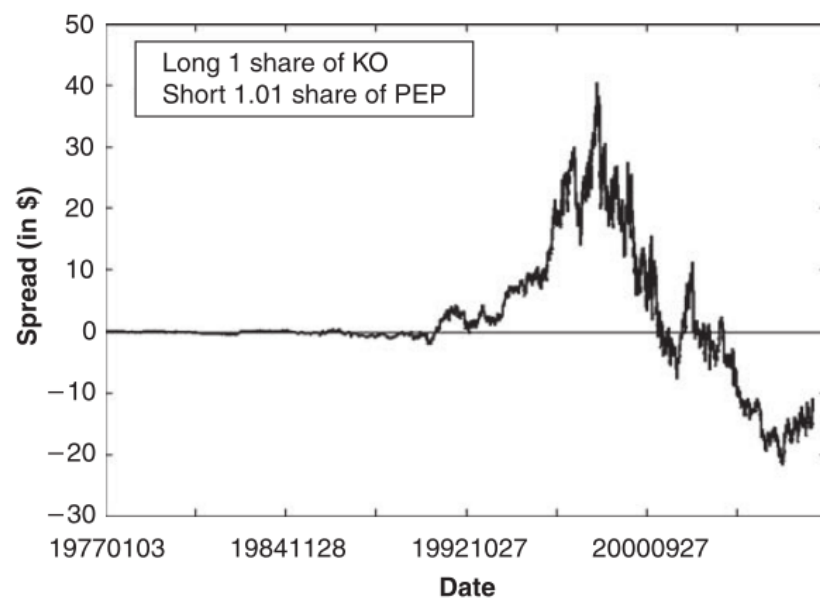


# 协整性

- 然而，如果两支股票是协整的，并且在未来仍然如此，它们的价格(适当加权)将不太可能出现分歧。
- 然而，它们每天(或每周，或任何其他时间段)的回报可能是完全不相关的。
- 适用均值回复策略

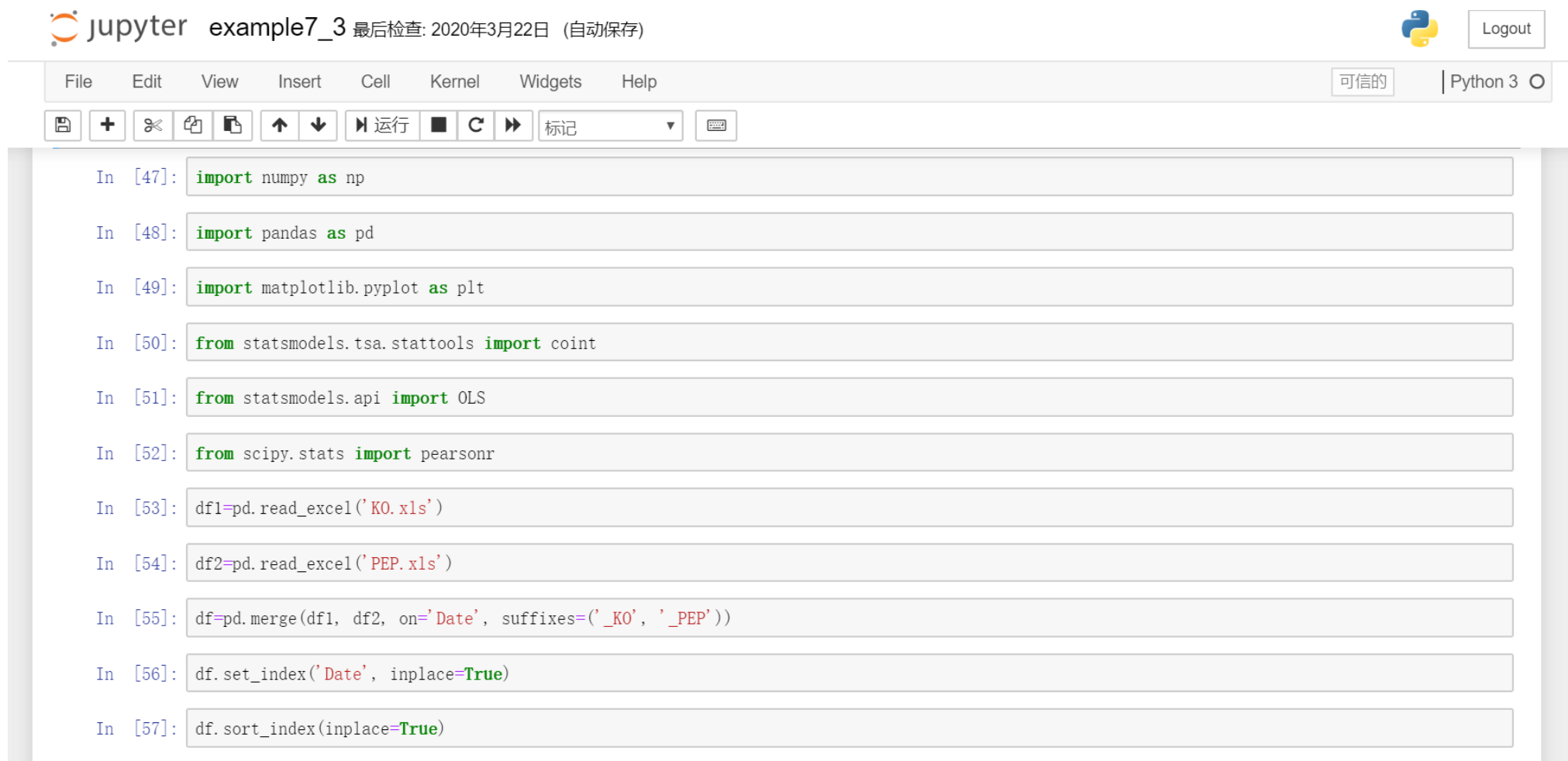
# 相关性 与协整性关系的示例：

- KO (Coca-Cola) versus PEP (Pepsi).
- 具有正相关性但不具有协整性



**FIGURE 7.5** A Nonstationary Time Series Formed by the Spread between KO and PEP

# KO和PEP的相关性和协整性测试的示例



The image shows a Jupyter Notebook interface with the title "example7\_3" and a timestamp "最后检查: 2020年3月22日 (自动保存)". The interface includes a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu bar is a toolbar with icons for saving, adding cells, undo, redo, and running code. The notebook contains ten code cells, each starting with "In [X]:" followed by Python code. The code imports numpy, pandas, and matplotlib, then imports specific functions from statsmodels and scipy. It reads two Excel files, "KO.xls" and "PEP.xls", merges them, sets the index to "Date", and sorts the index.

```
In [47]: import numpy as np

In [48]: import pandas as pd

In [49]: import matplotlib.pyplot as plt

In [50]: from statsmodels.tsa.stattools import coint

In [51]: from statsmodels.api import OLS

In [52]: from scipy.stats import pearsonr

In [53]: df1=pd.read_excel('KO.xls')

In [54]: df2=pd.read_excel('PEP.xls')

In [55]: df=pd.merge(df1, df2, on='Date', suffixes=('_KO', '_PEP'))

In [56]: df.set_index('Date', inplace=True)

In [57]: df.sort_index(inplace=True)
```

(example7\_3.ipynb)

# 协整性测试

## Run cointegration (Engle-Granger) test

```
In [58]: coint_t, pvalue, crit_value=coint(df['Adj Close_KO'], df['Adj Close_PEP'])
```

```
In [59]: (coint_t, pvalue, crit_value) # abs(t-stat) < critical value at 90%. pvalue says probability of null hypothesis (of no cointegration) is 73%
```

```
Out[59]: (-1.5815517041517158,  
          0.7286134576473535,  
          array([-3.89783854, -3.33691006, -3.04499143]))
```

结论：不协整

# 计算对冲比例

## Determine hedge ratio

```
In [60]: model=OLS(df['Adj Close_KO'], df['Adj Close_PEP'])
```

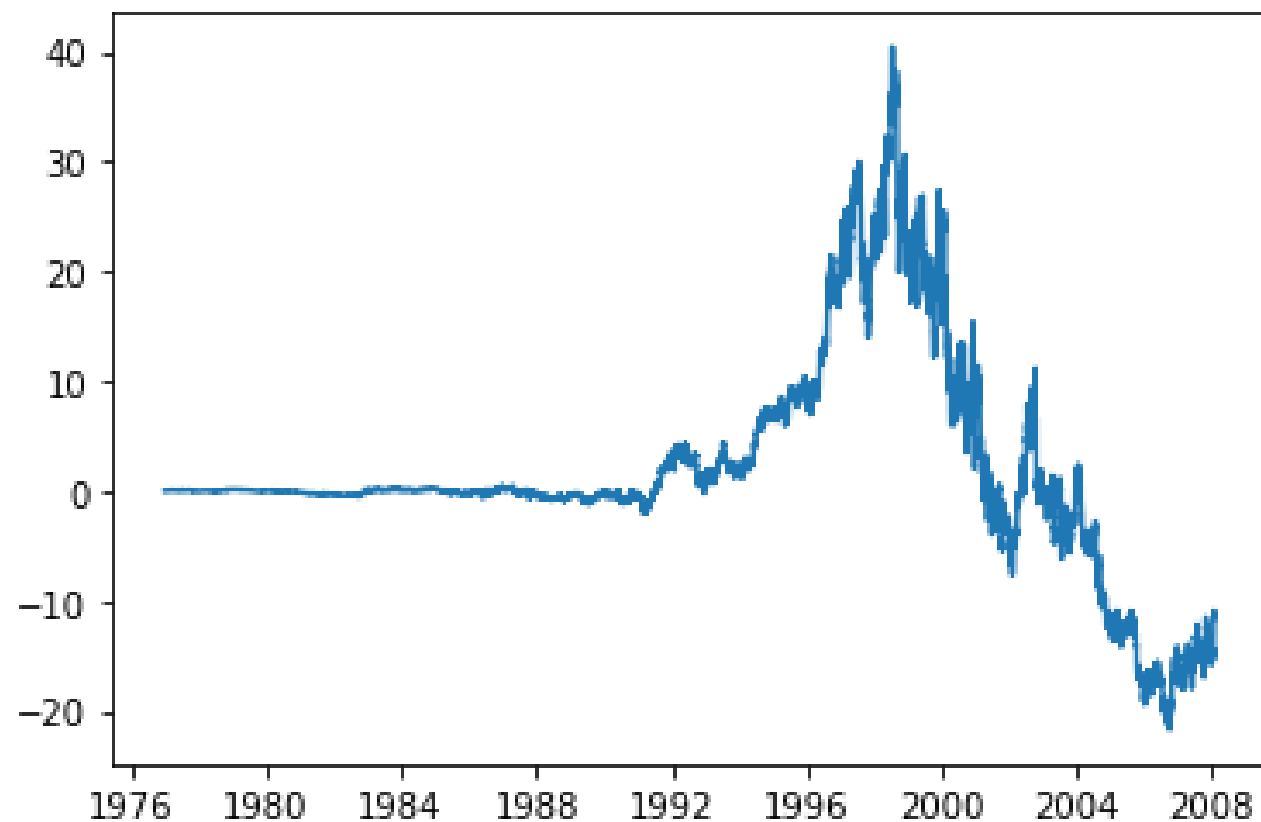
```
In [61]: results=model.fit()
```

```
In [62]: hedgeRatio=results.params
```

```
In [63]: hedgeRatio
```

```
Out[63]: Adj Close_PEP    1.011409  
dtype: float64
```

# 价格差 (Spread) 序列示意图



$\text{spread} = \text{KO} - \text{hedgeRatio} * \text{PEP}$

因两者之间无协整性，无法实现KO和PEP的配对交易

# 相关性测试

## Correlation test

```
In [66]: dailyret=df.loc[:, ('Adj Close_KO', 'Adj Close_PEP')].pct_change()
```

```
In [67]: dailyret.corr()
```

Out[67]:

	Adj Close_KO	Adj Close_PEP
Adj Close_KO	1.000000	0.484924
Adj Close_PEP	0.484924	1.000000

```
In [68]: dailyret_clean=dailyret.dropna()
```

```
In [69]: pearsonr(dailyret_clean.iloc[:,0], dailyret_clean.iloc[:,1]) # first output is correlation coefficient, second output is pvalue.
```

Out[69]: (0.48492394393705673, 0.0)

```
In [ ]:
```

结论：两者的相关性为0.4849

# 注意！！

- 收益率  $r_{t+1} = \frac{p_{t+1}}{p_t} - 1$ 。
- 其中p是收盘价，r收益率

1.相关性是指两只股票**收益率**的相关性。

2.协整性是指两只股票**价格**的多空组合是平稳的。



# 总结

- 配对交易的原理与步骤
- 相关性和协整性的概念
- 举了正反两个例子
  - GLD与GDX, 协整
  - KO和PEP, 相关, 不协整

# 参考资料

**[1] Ernest P. Chan, Quantitative Trading, Wiley 2008.**

谢谢指正！