

# 第七章 单位根检验和协整



# 学习目标与要求

- 了解伪回归现象的产生原因.
- 理解单位根检验的思想, 并掌握四种单位根检验方法.
- 理解协整的概念, 学会协整检验方法.
- 了解误差修正的思想.



# 本章结构

1. 伪回归

2. 单位根检验

3. 协整

4. 误差修正模型

# 伪回归

- 伪回归

随着时间序列分析的发展, 非平稳时间序列的理论逐步成为时间序列分析的主要分支之一, 并且在计量经济学的理论和实证分析中得到了广泛的应用. 从 20 世纪 70 年代开始, 人们逐渐发现用传统方法处理时间序列数据时, 经常会出现虚假回归现象, 也称为伪回归现象, 究其原因在于经典分析中隐含了许多假定条件, 如平稳性等. 如果将非平稳序列数据应用于传统的建模分析中, 实际上是默认了假定成立. 在这些假定条件成立的情况下, 传统的  $t$  检验和  $F$  检验具有可信度, 但是如果假定条件不满足, 那么传统的  $t$  检验和  $F$  检验是不可信的.



# “伪回归”现象

## ■ “伪回归”现象

考虑如下一元回归模型:  $y_t = \beta_0 + \beta_1 x_t + \xi_t$ . (7.1)

现在对该模型做显著性检验, 即

原假设  $H_0: \beta_1 = 0 \leftrightarrow$  备择假设  $H_1: \beta_1 \neq 0$ .

现在假定响应序列  $\{Y_t\}$  和 输入序列  $\{X_t\}$  相互独立, 就是说响应序列  $\{Y_t\}$  和 输入序列  $\{X_t\}$  之间没有显著的线性关系. 因此, 从理论上讲, 检验结果应该接受原假设  $\beta_1 = 0$ . 但是, 如果检验结果却接受备择假设, 即支持  $\beta_1 \neq 0$ . 那么就会得到响应序列  $\{Y_t\}$  和 输入序列  $\{X_t\}$  之间具有显著线性关系的错误结论, 从而承认

# “伪回归”现象

原本不成立的回归模型(7.1),这就犯了第一类错误,即拒真错误.

一般地,由于样本的随机性,拒真错误始终都会存在,不过通过设置显著性水平 $\alpha$ 可以控制犯拒真错误的概率. 构造  $t$  检验统计量:

$$t = \frac{\beta_1}{\sigma_{\beta_1}}.$$

当响应序列或输入序列都平稳时,该统计量服从自由度为样本容量  $n$  的  $t$  分布. 当  $|t| \leq t_{\alpha/2(n)}$  时,可以将拒真错误发生的概率控制在显著性水平 $\alpha$ 内,即



# “伪回归”现象

$$P_r(|t| \leq t_{\alpha/2(n)} | \text{平稳序列}) \leq \alpha$$

当响应序列和输入序列不平稳时, 检验统计量  $t$  不再服从  $t$  分布. 如果仍然采用  $t$  分布的临界值进行检验, 那么拒绝原假设的概率就会大大增加, 即

$$P_r(|t| \leq t_{\alpha/2(n)} | \text{非平稳序列}) \geq \alpha.$$

在这种情况下, 我们将无法控制拒真错误, 非常容易接受回归模型显著成立的错误结论, 这种现象称为“伪回归”现象或“虚假回归”现象.



# 非平稳对回归的影响

## ■ 非平稳对回归的影响

考虑如下回归问题：

假设 $\{X_t\}\{Y_t\}$ 是相互独立的随机游走序列，即

$$\begin{cases} x_t = x_{t-1} + u_t, u_t \sim N(\sigma_u^2); \\ y_t = y_{t-1} + v_t, v_t \sim N(\sigma_v^2), \end{cases}$$

式中 $\{u_t\}\{v_t\}$ 是相互独立的白噪声序列。现在形式地引入回归模型  $y_t = \alpha + \beta x_t + \varepsilon_t$ 。

由于序列 $\{X_t\}$ 与 $\{Y_t\}$ 不相关，所以 $\beta$ 应该为零。如果模拟结果显示拒绝原假设的概率远远大于显著性水平 $\alpha$





# 非平稳对回归的影响

那么我们认为伪回归显著成立.

Granger 和 Newbold 于 1974 年进行了蒙特卡罗 (Monte Carlo) 模拟. 模拟结果显示, 每 100 次回归拟合中, 平均有 76 次拒绝  $\beta_1 = 0$  的假定, 远远大于显著性水平  $\alpha = 0.05$ . 这说明在非平稳的场合, 参数显著性检验犯拒真错误的概率远大于  $\alpha$ , 即伪回归显著成立.

产生伪回归的原因是在非平稳场合, 参数  $t$  检验统计量不再服从  $t$  分布. 在样本容量  $n = 100$  的情况下, 进行大量的随机拟合, 得到  $\beta_1$  的  $t$  检验统计量的样本分布  $t(\hat{\beta}_1)$  的密度 (见图 7.1 中的虚线). 从图中可



# 非平稳对回归的影响

以看到,  $\beta_1$  的样本分布  $t(\hat{\beta}_1)$  的密度尾部肥, 方差大, 比  $t$  分布要扁平很多. 因此, 在  $t(\hat{\beta}_1)$  分布下,  $\hat{\beta}_1$  落入  $t$  分布所确定的显著性水平为 5% 的双侧拒绝域的概率远远大于 5% (见图 7.1)

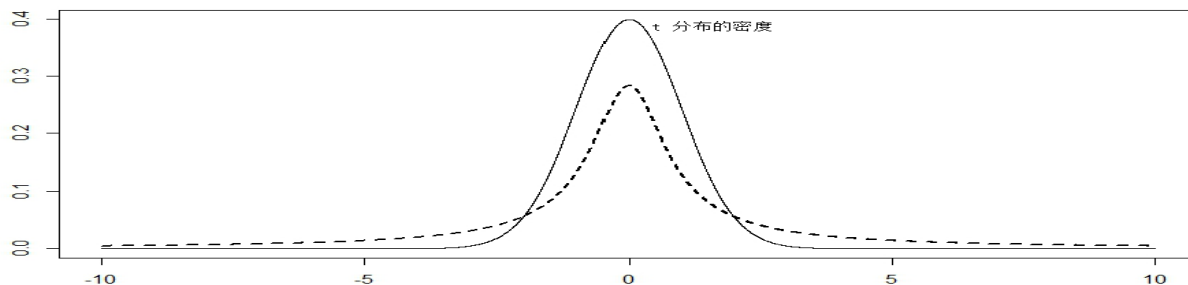


图 7.1 非平稳情况下参数检验统计量的样本拟合分布

# 本章结构

1. 伪回归

2. 单位根检验

3. 协整

4. 误差修正模型

# 单位根检验

由于伪回归问题的存在, 在进行回归建模时, 必须先检验各序列的平稳性. 只有各序列平稳了, 才可避开伪回归. 在前面我们主要是应用图检验的方法进行平稳性检验, 但是由于图检验带有很强的主观性, 因此必须研究平稳性的统计检验方法. 在实际问题中, 应用最广的是单位根检验. 从这一小节课开始, 我们来学习单位根检验方法.

## ■ 理论基础

首先考虑单位根序列, 这是一类最为常见的非平稳序列. 所谓单位根序列 (unit root series), 就是满足如下条件的序列  $\{X_t\}$  :



# 单位根检验

$$x_t = x_{t-1} + \xi_t, \quad (7.2)$$

其中,  $\{\xi_t\}$  为一平稳过程, 且

$E(\xi_t)=0, \text{COV}(\xi_t, \xi_{t-s})=\gamma(s)<\infty, s=0,1,2,\dots$  如果包含非零常数项:

$$x_t = \alpha + x_{t-1} + \xi_t, \quad (7.3)$$

那么称序列 $\{X_t\}$ 为带漂移的单位根序列 (unit root series with drift).

借助滞后算子, 将 (7.2) 式改写成如下形式:

$$(1-B)x_t = \xi_t.$$

滞后多项式的特征方程为 $1-B=0$ , 它的根为  $B=1$ . 这就是将序列 $\{X_t\}$ 称为单位根序列的原因. 当  $\xi_t$  为



# 单位根检验

白噪声序列  $\varepsilon_t$  时, (7.2) 式和 (7.3) 式表示的分别为随机游动序列和带漂移的随机游动序列.

若  $\{X_t\}$  为单位根序列, 则对其作 1 阶差分:  $\nabla x_t = \xi_t$ , 显然差分序列是一个平稳序列. 一般地, 我们把经过一次差分运算后变为平稳的序列称为一阶单整 (integration) 序列, 记为  $\{X_t\} \sim I(1)$ . 如果一个序列经过一次差分运算之后所得序列仍然非平稳, 而序列经过两次差分运算之后才变成平稳, 那么我们称该序列为二阶单整序列, 记为  $\{X_t\} \sim I(2)$ . 类似地, 如果一个序列经过  $n$  次差分之后平稳, 而  $n - 1$  次差分却不平稳, 那么称  $\{X_t\}$  为  $n$  阶单整序列, 记为  $\{X_t\} \sim I(n)$ .



# 单位根检验

有时我们也称平稳序列 $\{X_t\}$ 为零阶单整序列, 记为 $\{X_t\} \sim I(0)$ . 显然, 如果 $\{X_t\} \sim I(1)$ . 那么 $\{\nabla X_t\} \sim I(0)$ ; 如果 $\{X_t\} \sim I(2)$ . 那么 $\{\nabla^2 X_t\} \sim I(0), \dots$ .

单整衡量的是单个序列的平稳性, 它具有如下有用的性质:

- (1) 若 $\{X_t\} \sim I(0)$ , 则对于任意实数 $c_1, c_2$ , 有 $c_1 + c_2 x_t \sim I(0)$ .
- (2) 若 $\{X_t\} \sim I(d)$ , 则对于任意非零实数 $c_1, c_2$ , 有 $c_1 + c_2 x_t \sim I(d)$ .
- (3) 若 $\{X_t\} \sim I(0)$ , 则对于任意非零实数 $c_1, c_2$ , 有 $c_1 x_t + c_2 y_t \sim I(0)$ .
- (4) 若 $\{X_t\} \sim I(m), y_t \sim I(n)$ , 则对于任意非零实数 $c_1, c_2$ , 有 $c_1 x_t + c_2 y_t \sim I(k)$ , 其中 $k = \max\{m, n\}$ .



# 单位根检验

为了分析单位根序列, 我们简要介绍一些维纳过程和泛函中心极限定理的基本内容.

## 1. 维纳过程 (Wiener process)

设  $W(t)$  是定义在闭区间  $[0,1]$  上的连续变化的随机过程, 若该过程满足:

(1)  $W(0) = 0$ ;

(2) 独立增量过程: 对闭区间  $[0,1]$  上任意一组分割  $0 \leq t_1 < t_2 < \cdots < t_k = 1$ ,  $W(t)$  的增量:

$$W(t_2) - W(t_1), W(t_3) - W(t_2), \cdots, W(t_k) - W(t_{k-1})$$

为相互独立的随机变量;

(3) 对任意  $0 \leq s < t \leq 1$ , 有  $W(t) - W(s) \sim N(0, t-s)$ .





# 单位根检验

则称 $W(t)$ 为标准维纳过程,也称为布朗运动(Brownian motion).

由定义可见,标准维纳过程是一个正态独立增量过程,且 $W(t_2)-W(t_1), W(t)=W(t)-W(0) \sim N(0,t), W(1) \sim N(0,1)$ .

将标准维纳过程 $W(t)$ 推广,可得到一般维纳过程.

令 $B(t)=\sigma W(t)$ ,则 $B(t)$ 是一个方差为 $\sigma^2$ 的维纳过程,即 $B(t)$ 满足维纳过程定义中的(1),(2)两条,以及对任意 $0 \leq s < t \leq 1$ ,有 $B(t)-B(s) \sim N(0, \sigma^2(t-s))$ ,

从而 $B(t)=B(t)-B(0) \sim N(0, \sigma^2 t), B(1) \sim N(0, \sigma^2)$ .

维纳过程在理论和实践中都有重要作用,比如用它可



# 单位根检验

构造其他的过程: 令  $y_t = [W(t)]^2$  则  
$$y_t \sim t_{\chi^2}(1), \forall t > 0.$$

## 2. 泛函中心极限定理

泛函中心极限定理是研究非平稳时间序列的重要工具, 具体内容如下:

设序列  $\{\varepsilon_t\}$  满足条件:  $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_t, \dots$  独立同分布, 且

$$E(\varepsilon_t) = 0, D(\varepsilon_t) = \sigma^2, t = 1, 2, \dots$$

$w$  为闭区间  $[0, 1]$  上的任一实数, 给定样本  $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N$ , 记  $M_w = [wN]$ , 则当  $N \rightarrow \infty$  时,

$$\frac{1}{\sqrt{N}} \sum_{t=1}^{M_w} \varepsilon_t \xrightarrow{L} B(w) = \sigma W(w).$$



# DF 检验

在上式中令  $w=1$ , 有

$$\frac{1}{\sqrt{N}} \sum_{t=1}^M \varepsilon_t \xrightarrow{L} B(1) = \sigma W(1) \sim N(0, \sigma^2).$$

该定理是单位根检验统计量极限分布推导的理论基础, 许多单位根检验方法统计量的极限分布都是基于此定理构造出来的.

## ■ DF 检验

非平稳序列构成复杂, 不同结构的非平稳序列分析方法也不尽相同. 在进行单位根检验时, 通常把非平稳分成如下三种类型.

### 1. 无漂移项自回归情形



# DF 检验

考虑如下的简单 AR(1) 模型:  $x_t = \phi x_{t-1} + \varepsilon_t$ , (7.4)

其中,  $\varepsilon_t$  独立同分布, 且  $\varepsilon_t \sim N(0, \sigma^2)$ . 单位根检验的假设为:

原假设  $H_0: \phi = 1 \Leftrightarrow x_t \sim I(1) \leftrightarrow$  备择假设  $H_1: |\phi| < 1 \Leftrightarrow x_t \sim I(0)$ .

构造检验统计量 
$$t = \frac{\hat{\rho} - 1}{\hat{\sigma}_{\hat{\rho}}},$$

其中,  $\hat{\rho}$  为 AR(1) 模型中的最小二乘估计量,  $\hat{\sigma}_{\hat{\rho}}$  为估计量  $\hat{\rho}$  的标准差. 可以证明, 在原假设成立的情况下,  $t$  统计量依分布收敛于维纳过程的泛函, 即

$$t \xrightarrow{L} \frac{1}{2} \left[ W^2(1) - 1 \right] / \left[ \int_0^1 W^2(r) dr \right]^{\frac{1}{2}}.$$



# DF 检验

这说明  $t$  检验统计量不再服从传统的  $t$  分布, 传统的  $t$  检验法失效. 上面的极限分布一般称为Dickey-Fuller 分布, 对应的检验称为 DF 检验. 进一步的考察表明, 上述  $t$  检验统计量的极限分布是非对称、左偏的. 又因为  $x_t = \alpha + \phi x_{t-1} + \varepsilon_t$ , 所以检验值大都是负数.

对于 Dickey-Fuller 分布, 可用蒙特卡罗方法模拟得到检验的临界值, 并编成 DF 临界值表供查询. 在进行 DF 检验时, 比较  $t$  统计量与 DF 检验临界值, 就可在某个显著性水平上拒绝或接受原假设. 若  $t$  统计量小于 DF 检验临界值, 则拒绝原假设, 说明序列不存在单位根; 若  $t$  统计量值大于或等于 DF 检验临界值, 则接受原假设, 说明序列存在单位根. 图 7.2 为以样本容量为 100, 进行 10000 次模拟, 得到模拟的  $t$



# DF 检验

统计量分布的密度函数图. 从图中可以看到, 相对于正态分布的密度,  $t$  统计量分布的密度呈现出左偏分布.

## 2. 带漂移项自回归情形

考虑如下带漂移项的自回归模型:  $x_t = \alpha + \phi x_{t-1} + \varepsilon_t$ ,

其中, 引入  $\alpha$  是为了捕捉非零均值. 作如下检验假设:

原假设  $H_0: \phi = 1 \Leftrightarrow x_t \sim I(1) \leftrightarrow$  备择假设  $H_1: |\phi| < 1 \Leftrightarrow x_t \sim I(0)$ .

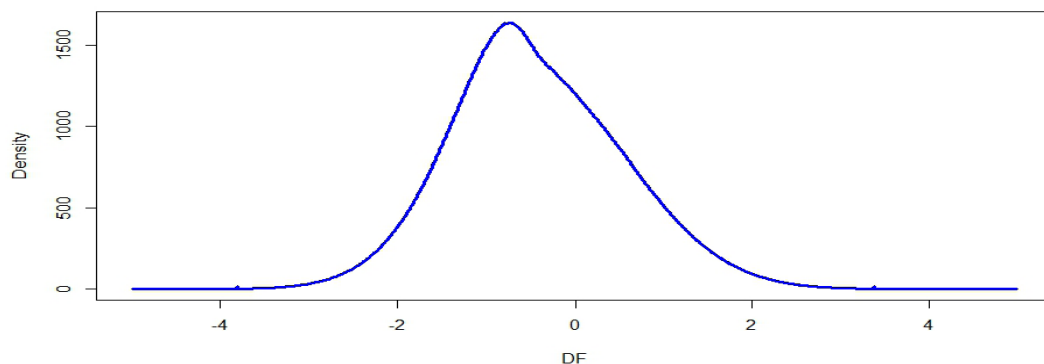


图 7.2 DF 检验的  $t$  统计量密度图

# DF 检验

在原假设条件下, 检验的  $t$  统计量的渐近分布的分位数和  $\rho$  值都可以通过常用软件进行计算. 由于在这种情况下,  $t$  统计量的渐近分布有些繁复, 所以这里就不再列出.

在 R 语言中, 可以用函数 `qunitroot()` 来计算单位根检验  $t$  统计量的样本分位数. 在使用该函数之前, 需要调用程序包 `urca`. 函数 `qunitroot()` 的命令格式如下:

```
qunitroot(p, N= ,trend= ,statistic =)
```

该函数的参数说明:

- `p`: 概率值向量.
- `N`: 样本观察值的个数.
- `trend`: `trend` 通常可取三个值, `trend= "c"` 意味着模型



# DF 检验

中不含漂移项; trend= “ct” 意味着模型中含有漂移项和时间趋势项. 默认取值为 c.

- statistic: statistic 仅取两个值. statistic= “t” 表示检验用 t 统计量; statistic= “n” 表示检验用标准化统计量, 有时称为  $\rho$  统计量. 默认值取为 t.

例 7.1 分别计算没有漂移项和含有漂移项的单位根检验下, t 统计量的概率值分别为 0.25, 0.5, 0.75, 0.85 的分位数.

解 基于 MacKinnon (1996) 的研究, 使用函数 `qunitroot()` 进行计算. 具体命令及运行结果如下:

```
>qunitroot(c(0.25,0.5,0.75,0.85),N=100,trend="nc",statistic ="t")
```

```
[1] -1.0853871 -0.4935008 0.2264920 0.6262640
```





# DF 检验

```
>qunitroot(c(0.25,0.5,0.75,0.85),N=100,trend="c",statistic ="t")
```

```
[1] -2.0879152 -1.5584449 -1.0028885 -0.6640883
```

可见, 模型中含有漂移项的  $t$  统计量比不含漂移项的  $t$  统计量分布更偏左.

## 3. 带趋势自回归情形

考虑如下回归模型:  $x_t = \alpha + \beta_t + \phi x_{t-1} + \varepsilon_t$ ,

其中, 引入常数项  $\alpha$  和时间趋势  $\beta_t$  是为了捕捉确定性趋势. 这类型适合带有趋势性的时间序列, 像资产价格或者宏观总量指标等. 类似地, 作如下检验假设:

原假设  $H_0: \phi = 1 \Leftrightarrow x_t \sim I(1) \Leftrightarrow$  备择假设  $H_1: |\phi| < 1 \Leftrightarrow x_t \sim I(0)$ .

在原假设条件下, 检验的  $t$  统计量的渐近分布受到常数项



# DF 检验

和时间趋势项的影响, 其分位数和 $\rho$  值也都可以通过常用软件进行计算.

在 R 语言中, `fUnitRoots` 程序包中的函数 `adfTest()` 和 `unitrootTest()` 都可以进行单位根检验. `tseries` 程序包中的函数 `adf.test()` 和 `pp.test()` 也可以方便地进行单位根检验. `urca` 程序包中的函数 `ur.df()` 和 `ur.pp()` 也都可以快捷地进行单位根检验. 函数 `adfTest()` 的命令格式如下:

```
adf.test(x, alternative= ,k= )
```

该函数的参数说明:

- `x`: 需要进行单位根检验的序列名.
- `alternative`: 表明备择假设. `alternative= "stationary"`



# DF 检验

“explosive” .

- k: 延迟阶数. k 默认取值为  $\text{trunc}((\text{length}(x)-1)^{\frac{1}{3}})$  至于 `unitrootTest()`、`pp.test()` 等函数的用法类似.

例 7.2 分析 1996 年至 2015 年国内居民出境人数的对数序列  $\{\ln x_t\}$  和外国人入境游客人数的对数序列  $\{\ln y_t\}$  并进行 DF 检验. (单位: 万人次)

解 首先绘制两个序列的时序图. 然后分别按三种模型类型进行检验. 具体命令如下, 运行结果如图 7.3 所示.

```
> ys <- read.csv("E:/DATA/CHAP7/7.2.csv",header=T)
> xt <- ts(log(ys$GNJMCG),start=1996)
> yt <- ts(log(ys$WGRRJ),start=1996)
> l1 <- min(xt,yt)
> l2 <- max(xt,yt)
> plot(xt,ylim=c(l1,l2),type="o",lwd=2,col=2,pch=20,ylab="xt-yt")
```



# DF 检验

```
> lines(yt,lty=2,col=4,type="o",pch=20)  
> legend(x="topleft",c("xt","yt"),lwd=c(2,1),lty=c(1,2),col=c(2,4))
```

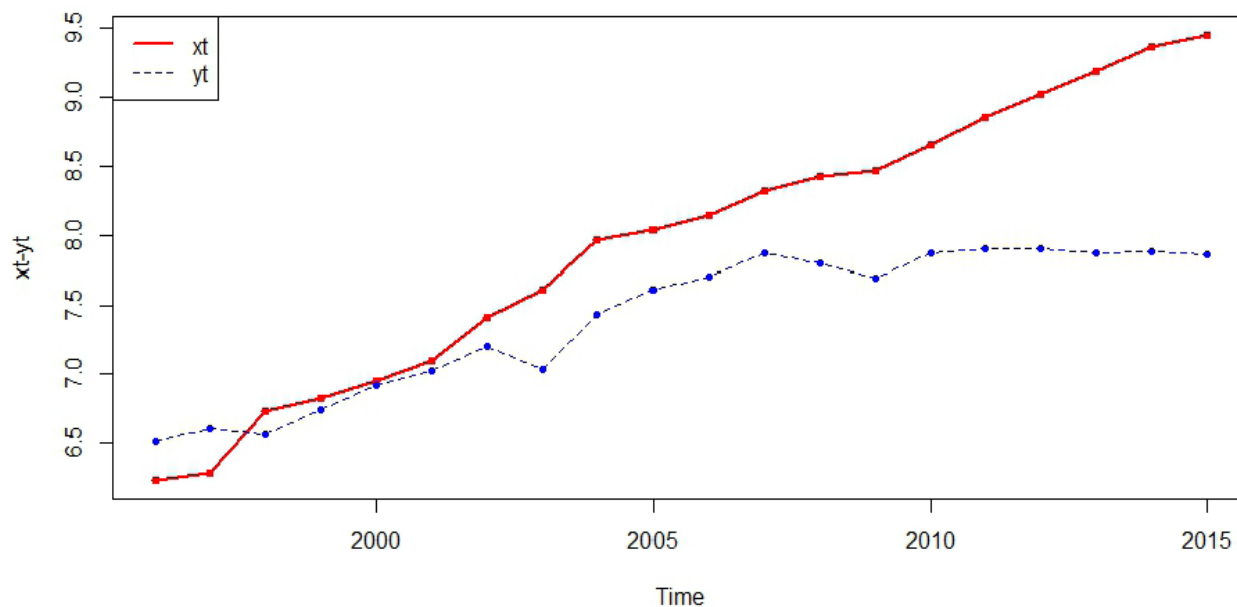


图 7.3 国内居民出境人数的对数序列  $\{\ln x_t\}$  和外国人入境游客人数的对数序列  $\{\ln y_t\}$  的时序图

# DF 检验

对国内居民出境人数的对数序列  $\{\ln x_t\}$  序列进行 DF 检验。  
具体命令及运行结果如下：

```
> adfTest(xt,lag=1,type="nc")
```

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: 3.7254

P VALUE:

0.99

```
> adfTest(xt,lag=1,type="c")
```

Title:

Augmented Dickey-Fuller Test



# DF 检验

Test Results:

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: -2.0516

P VALUE:

0.3047

`adfTest(xt,lag=1,type="ct")`

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: -2.309

P VALUE:



# DF 检验

0.4547

```
> adf.test(xt,k=1)
```

Augmented Dickey-Fuller Test

data: xt

Dickey-Fuller = -2.309, Lag order = 1, p-value = 0.4547

alternative hypothesis: stationary

检验结果表明,  $\{\ln x_t\}$  序列滞后 1 阶的自回归系数为 1, 且更倾向于满足无漂移的自回归模型类型. 对外国人入境游客人数的对数序列  $\{\ln y_t\}$  进行 DF 检验. 具体命令及运行结果如下:



# DF 检验

```
> adfTest(yt,lag=1,type="nc")
```

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: 1.9688

P VALUE:





# DF 检验

0.9838

```
> adfTest(yt,lag=1,type="c")
```

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: -1.7131

P VALUE:



# DF 检验

0.4246

```
> adfTest(yt,lag=1,type="ct")
```

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: -0.5394

P VALUE:

0.9717



# ADF 检验

0.9717

```
> adf.test(yt,k=1)
```

Augmented Dickey-Fuller Test

data: yt

Dickey-Fuller = -0.53936, Lag order = 1, p-value = 0.9717

alternative hypothesis: stationary

检验结果表明  $\{y_t\}$  序列滞后 1 阶的自回归系数为 1, 且更倾向于满足无漂移的自回归模型类型.

## ■ ADF 检验

单位根 DF 检验只适用于带有白噪声的 AR(1) 模型, 然而许多金融时间序列可能包含更为复杂的动态结构, 并不能用



# ADF 检验

baixd\_dlun@16

简单的AR(1) 模型来刻画. 为了检验高阶的自回归模型<sup>from</sup> AR(p),  $p > 1$ , Dickey 和 Fuller 1979 年提出了增广 DF (augmented Dickey-Fuller) 检验, 简称为 ADF 检验.

## 1. ADF 检验的原理

假设时间序列  $\{x_t\}$  服从 AR(p) 过程:

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \varepsilon_t, \quad (7.5)$$

其中,  $\varepsilon_t$  为白噪声. 将 (7.5) 式变形为

$$\begin{aligned} x_t &= [\phi_1 x_{t-1} + (\phi_2 + \phi_3 + \cdots + \phi_p) x_{t-1}] \\ &\quad - [(\phi_2 + \phi_3 + \cdots + \phi_p) x_{t-1} - \phi_2 x_{t-2} - (\phi_3 + \phi_4 + \cdots + \phi_p) x_{t-2}] \\ &\quad - (\phi_3 + \cdots + \phi_p) x_{t-2} + \cdots - \phi_p x_{t-p+1} + \phi_p x_{t-p} + \varepsilon_t \\ &= (\phi_1 + \phi_2 + \cdots + \phi_p) x_{t-1} - (\phi_2 + \phi_3 + \cdots + \phi_p) \nabla x_{t-1} - \cdots - \phi_p \nabla x_{t-p+1} + \varepsilon_t \\ &= \alpha x_{t-1} + \beta_1 \nabla x_{t-1} + \beta_2 \nabla x_{t-2} + \cdots + \beta_{p-1} \nabla x_{t-p+1} + \varepsilon_t \end{aligned} \quad (7.6)$$



# ADF 检验

其中,  $\alpha = \phi_1 + \phi_2 + \cdots + \phi_p$ ,  $\beta_i = -(\phi_{i+1} + \cdots + \phi_p)$ ,  $i = 1, 2, \cdots, p-1$ .  
如果将  $p-1$  个滞后项  $\nabla x_{t-i}$  ( $i = 1, 2, \cdots, p-1$ ) 归到随机干扰项中, 则干扰项就成为序列相关的平稳过程. 于是, 将 (7.6) 式与 (7.4) 式相比较, 对模型 (7.6) 的单位根检验就是对干扰项为一平稳过程的单位根检验.

通过上面分析可知, AR(p) 模型单位根检验的假设条件为

原假设  $H_0: \alpha = 1$  (序列  $\{x_t\}$  非平稳)  $\longleftrightarrow$  备择假设  $H_1: \rho < 0$  (序列  $\{x_t\}$  平稳).

构造 ADF 检验统计量:

$$\tau = \hat{\rho} / \sigma_{\hat{\rho}},$$



# ADF 检验

其中,  $\hat{\rho}$  为原假设条件下, 对模型 (7.7) 进行的最小二乘估计.  
 $\hat{\rho}$  为参数  $\rho$  的样本标准差.

通过蒙特卡罗方法, 可以得到  $\tau$  检验统计量的临界值表. 显然 DF 检验是 ADF 检验在自相关阶数为 1 时的一个特例, 所以统称为 ADF 检验.

## 2. ADF 检验的类型

在实际应用中, 和 DF 检验一样, ADF 检验也可以用于如下三种类型的单位根检验:

(1) 无漂移项、无时间趋势项的  $p$  阶自回归模型:

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \varepsilon_t.$$



# ADF 检验

(2) 有漂移项、无时间趋势项的  $p$  阶自回归模型:

$$x_t = \mu + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \varepsilon_t.$$

(3) 有漂移项、有时间趋势项的  $p$  阶自回归模型:

$$x_t = \mu + \beta t + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \varepsilon_t.$$

相应地, 我们可得 ADF 检验回归方程 (1), (2), (3) 的另一种形式分别为 (7.7) 式和

$$\nabla x_t = \mu + \rho x_{t-1} + \beta_1 \nabla x_{t-1} + \beta_2 \nabla x_{t-2} + \cdots + \beta_{p-1} \nabla x_{t-p+1} + \varepsilon_t,$$

$$\nabla x_t = \mu + \beta t + \rho x_{t-1} + \beta_1 \nabla x_{t-1} + \beta_2 \nabla x_{t-2} + \cdots + \beta_{p-1} \nabla x_{t-p+1} + \varepsilon_t.$$

例 7.3 对 1996 年至 2015 年国内居民出境人数的对数差分序列  $\{\nabla \ln x_t\}$  和外国人入境游客人数的对数差分后序列  $\{\nabla \ln x_t\}$  进行 ADF 检验.



# ADF 检验

解 首先加载程序包 fUnitRoots; 然后读入数据, 作取对数处理, 建立时间序列, 并做差分; 最后分别做三种类型的 ADF 检验. 具体命令及运行结果如下:

```
> library(fUnitRoots)

> ys <- read.csv("E:/DATA/CHAP7/7.2.csv",header=T)

> xt <- ts(log(ys$GNJMCg),start=1996)

> yt <- ts(log(ys$WGRRJ),start=1996)

> dx <- diff(xt)

> dy <- diff(yt)

> for(i in 1:3)print(adfTest(dx,lag=i,type="nc"))
```

Title:





# ADF 检验

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: -1.5838

P VALUE:

0.1051

Title:

Augmented Dickey-Fuller Test



# ADF 检验

Test Results:

PARAMETER:

Lag Order: 2

STATISTIC:

Dickey-Fuller: -0.8501

P VALUE:

0.3381

Title:

Augmented Dickey-Fuller Test

Test Results:



# ADF 检验

PARAMETER:

Lag Order: 3

STATISTIC:

Dickey-Fuller: -0.4841

P VALUE:

0.4542

```
> for(i in 1:3)print(adfTest(dx,lag=i,type="c"))
```

Title:

Augmented Dickey-Fuller Test

Test Results:



# ADF 检验

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: -3.362

P VALUE:

0.02386

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:



# ADF 检验

Lag Order: 2

STATISTIC:

Dickey-Fuller: -2.4831

P VALUE:

0.152

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 3



# ADF 检验

STATISTIC:

Dickey-Fuller: -1.5123

P VALUE:

0.4957

```
> for(i in 1:3)print(adfTest(dx,lag=i,type="ct"))
```

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 1



# ADF 检验

STATISTIC:

Dickey-Fuller: -3.2992

P VALUE:

0.09178

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 2

STATISTIC:



# ADF 检验

Dickey-Fuller: -2.8003

P VALUE:

0.2675

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 3

STATISTIC:

Dickey-Fuller: -2.0453





# ADF 检验

P VALUE:

0.5551

```
> for(i in 1:3)print(adfTest(dy,lag=i,type="nc"))
```

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: -1.9733



# ADF 检验

P VALUE:

0.04812

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 2

STATISTIC:

Dickey-Fuller: -1.3116

P VALUE:



# ADF 检验

0.1915

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 3

STATISTIC:

Dickey-Fuller: -1.3472

P VALUE:

0.1803



# ADF 检验

```
> for(i in 1:3)print(adfTest(dy,lag=i,type="c"))
```

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: -2.8038

P VALUE:

0.07651



# ADF 检验

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 2

STATISTIC:

Dickey-Fuller: -1.6816

P VALUE:

0.4357

Title:



# ADF 检验

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 3

STATISTIC:

Dickey-Fuller: -1.5607

P VALUE:

0.4785

```
for(i in 1:3)print(adfTest(dy,lag=i,type="ct"))
```

Title:



# ADF 检验

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: -3.921

P VALUE:

0.02707

Title:

Augmented Dickey-Fuller Test



# ADF 检验

Test Results:

PARAMETER:

Lag Order: 2

STATISTIC:

Dickey-Fuller: -2.8665

P VALUE:

0.2423

Title:

Augmented Dickey-Fuller Test

Test Results:





# ADF 检验

PARAMETER:

Lag Order: 3

STATISTIC:

Dickey-Fuller: -2.7895

P VALUE:

0.2716

检验结果表明, 国内居民出境人数的对数差分序列是有漂移项的平稳序列, 该平稳序列 1 阶自相关 (ADF 检验 p 值等于 0.02386). 外国人入境游客人数的对数差分序列一个是有漂移项和时间趋势项的 1 阶自相关模型 (ADF 检验 p 值等于 0.02707).



# PP 单位根检验

## ■ PP 单位根检验

Phillips 和 Perron 1988 年提出了一种单位根检验的方法, 简称为 PP 检验. 这种方法在时间序列分析中得到了广泛使用. PP 检验的回归方程如下:

$$\nabla x_t = \beta D_t + \rho x_{t-1} + \mu_t,$$

其中,  $D_t$  包含确定性成分, 如常数项或常数项加时间趋势项;  $\mu_t$  是一个 0 阶单整序列. PP 检验与 ADF 检验最大的不同在于, PP 检验可以通过调整统计量, 使其能够适用于任何存在序列相关和异方差的误差项  $\mu_t$ .

在原假设  $\rho = 0$  的条件下, PP 检验的 Z 统计量与 ADF 检验的  $\tau$  统计量具有相同的渐近分布, 因而具有相同的临界值.



# PP 单位根检验

在 R 语言中, urca 程序包中的函数 `ur.pp()` 可以进行 PP 单位根检验. 函数 `ur.pp()` 的命令格式如下:

```
ur.pp(x, type= ,model= ,lags= , use.lag= )
```

该函数的参数说明:

- x: 需要进行 PP 单位根检验的序列名.
- type: 检验统计量类型. type 可以取 “Z-alpha” 和 “Z-tau” 两个值.
- model: 回归模型的类型. model 通常可取两个值, model= “constant” 意味着模型中含有漂移项, 而无时间趋势; model= “trend” 意味着模型中不含漂移项, 而含时间趋势项.



# PP 单位根检验

- lags: 滞后阶数. 可取两个值: lags = “short” 或 “long”, 表示两个特定阶数.
- use.lag: 使用者自己指定的阶数.

例 7.4 对 1996 年至 2015 年国内居民出境人数的对数序列  $\{\ln x_t\}$  和外国人入境游客人数的对数序列  $\{\ln y_t\}$ , 进行 PP 检验.

解 读入数据, 并进行 PP 检验. 具体命令及运行结果如下:

```
> ys <- read.csv("E:/DATA/CHAP7/7.2.csv",header=T)
> xt <- ts(log(ys$GNJMCG),start=1996)
> yt <- ts(log(ys$WGRRJ),start=1996)
> gx1.pp <- ur.pp(as.vector(xt),type="Z-tau",model="constant",
```



# PP 单位根检验

```
+ lags="long")
```

```
> gx1.pp@teststat
```

```
[1] -2.219878
```

```
> gx1.pp@cval
```

```
1pct 5pct 10pct
```

```
critical values -3.830262 -3.029363 -2.655194
```

```
> gy1.pp <- ur.pp(as.vector(yt),type="Z-tau",model="trend",
```

```
+ lags="long")
```

```
> gy1.pp@teststat
```

```
[1] -0.7135191
```



# PP 单位根检验

```
> gy1.pp@cval
```

```
1pct 5pct 10pct
```

```
critical values -4.534844 -3.67457 -3.27616
```

可以看到, 两个序列的  $Z_\tau$  统计量值分别为  $-2.219878$  和  $-0.7135191$ , 都大于各自 1%、5%和 10% 显著性水平下的临界值, 因此不能够拒绝存在单位根的原假设. 进一步对两个序列差分分别作 PP 检验. 具体命令及运行结果如下:

```
> gx2.pp <- ur.pp(as.vector(diff(xt)),type="Z-tau",model="constant",
```

```
+ lags="long")
```

```
> gx2.pp@teststat
```

```
[1] -5.39613
```



# PP 单位根检验

```
> gx2.pp@cval
```

	1pct	5pct	10pct
--	------	------	-------

critical values	-3.857056	-3.040014	-2.660816
-----------------	-----------	-----------	-----------

```
>
```

```
> gy2.pp <- ur.pp(as.vector(diff(yt)),type="Z-tau",model="trend",  
+ lags="long")
```

```
> gy2.pp@teststat
```

```
[1] -5.784781
```

```
> gy2.pp@cval
```

	1pct	5pct	10pct
--	------	------	-------

critical values	-4.574275	-3.69202	-3.285628
-----------------	-----------	----------	-----------



# KPSS 单位根检验

可见, 两个序列的对数差分序列的  $Z_\tau$  统计量值分别为  $-5.39613$  和  $-5.784781$ , 都小于各自 1%、5% 和 10% 显著性水平下的临界值, 因此拒绝存在单位根的原假设, 有理由认为差分序列平稳. 这一结果也与例 7.3 的结论一致.

## ■ KPSS 单位根检验

Kwiatkowski、Phillips、Schmidt 和 Shin 1992 年提出了所谓的 KPSS 检验, 用于序列平稳性的检验. 设时间序列  $\{x_t, 0 < t < T\}$  满足方程

$$x_t = \beta D_t + w_t + u_t,$$

$$w_t = w_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim WN(0, \sigma_\varepsilon^2),$$





# KPSS 单位根检验

其中,  $D_t$  包含确定性成分;  $u_t$  是一个可能含有异方差的平稳序列;  $w_t$  是一个随机游走过程.

KPSS 检验的原假设是序列  $\{x_t\}$  平稳, 即  $\sigma_\varepsilon^2 = 0$ ; 备择假设是序列  $\{x_t\}$  非平稳, 即  $\sigma_\varepsilon^2 > 0$ . KPSS 检验的统计量为

$$\text{KPSS} = \frac{\sum_{t=1}^T s_t^2}{T^2 \hat{\sigma}_\varepsilon^2}$$

其中,  $s_t = u_1 + u_2 + \cdots + u_t, t = 1, 2, \cdots, T, u_t$  为  $x_t$  对  $D_t$  回归后的残差,  $\hat{\sigma}_\varepsilon^2$  是  $u_t$  的长期方差的一致估计量. 在序列平稳的原假设下, 当  $D_t$  恒为常数时, 有

$$\text{KPSS} \xrightarrow{L} \int_0^1 V_1(r) dr, \quad (7.8)$$



# KPSS 单位根检验

其中,  $V_1(r) = W(r) - rW(1)$ ,  $W(r)$  是一个标准布朗运动. 当  $D_t$  中含有时间趋势时, 有

$$\text{KPSS} \xrightarrow{L} \int_0^1 V_2(r) dr, \quad (7.9)$$

其中,  $V_2(r) = W(r) + r(2-3r)W(1) + 6r(r^2-1)\int_0^1 W(s)ds$ .

KPSS 检验的临界值可以通过模拟得出. 平稳性检验是一个单边右尾检验, 如果 KPSS 检验统计量大于 (7.8) 式或 (7.9) 式的  $1-\alpha$  分位数, 我们可以拒绝序列为平稳的假设.

在 R 语言中, urca 程序包中的函数 `ur.kpss()` 可以进行 KPSS 检验. 函数 `ur.kpss()` 的命令格式如下:

```
ur.kpss(x, type= ,lags= , use.lag= )
```



# KPSS 单位根检验

该函数的参数说明:

- `x`: 需要进行 KPSS 检验的序列名.
- `type`: 检验模型的类型. `type` 可以取 “mu” 和 “tau” 两个值. `type= “mu”` 意味着检验模型中含有常数项; `type= “tau”` 意味着检验模型中含有常数项和时间趋势项.
- `lags`: 滞后阶数. 可取三个值: `lags = “short”` 或 “long”, 表示两个特定阶数; `lags= “nil”` 表示无误差修正.
- `use.lag`: 使用者自己指定的阶数.

此外, 也可用程序包 `tseries` 中的函数 `kpss.test()` 来进行 KPSS 检验.



# KPSS 单位根检验

例 7.5 使用 KPSS 检验法检验 2014 年 10 月至 2017 年 8 月我国货币月度供应量 (期末值) 的对数差分序列的平稳性.

解 读入数据, 并取对数, 然后对序列对数差分作 KPSS 检验. 具体命令及运行结果如下:

```
> gy <- read.csv("E:/DATA/CHAP7/7.5.csv",header=T)
> gy.log.ts <- ts(log(gy$GYL),start=c(2014,10),frequency = 12)
> gy.kpss <- ur.kpss(diff(gy.log.ts),type="mu",use.lag=3)
> summary(gy.kpss)

#####

# KPSS Unit Root Test #

#####
```



# 本章结构

1. 伪回归

2. 单位根检验

3. 协整

4. 误差修正模型

# 协整

## ■ 协整

在 7.1 节中, 我们讨论了伪回归现象, 其实避免伪回归的方法很多, 其中一种办法是避免回归方程中出现非平稳时间序列

变量. 一般的做法是对非平稳时间序列变量进行差分, 使得差分序列变成平稳序列, 然后对差分变量进行回归. 虽然这种做法可以消除因变量非平稳带来的伪回归, 但是差分运算会损失变量的部分信息, 尤其是多次差分运算损失的信息量更大. 这是需要格外引起注意的问题. 另一种办法是直接对非平稳变量进行回归, 寻找变量之间存在的相依关系, 以建立反映水平变量间长期关系的回归方程. 这种方法就是本节介绍的协整分析.



# 协整

变量. 一般的做法是对非平稳时间序列变量进行差分, 使得差分序列变成平稳序列, 然后对差分变量进行回归. 虽然这种做法可以消除因变量非平稳带来的伪回归, 但是差分运算会损失变量的部分信息, 尤其是多次差分运算损失的信息量更大. 这是需要格外引起注意的问题. 另一种办法是直接对非平稳变量进行回归, 寻找变量之间存在的相依关系, 以建立反映水平变量间长期关系的回归方程. 这种方法就是本节介绍的协整分析.

## ■ 协整的概念

在现实生活中, 有些序列自身的变化虽然是非平稳的, 但是序列与序列之间却具有非常密切的长期均衡关系. 例如: 收入与消费、工资与价格、政府支出与税收、出口与进口, 等等, 这



# 协整

些经济时间序列一般各自都是非平稳的序列,但是它们之间却往往存在着长期均衡关系. 这些均衡关系具体表现为它们的某个线性组合保持着长期稳定的关系,这种现象就是所谓的协整(cointegration).

为进一步认识协整,我们看一个简单的模拟系统.

例 7.6 已知时间序列  $\{x_{1t}\}, \{x_{2t}\}$  满足如下系统:

$$\begin{cases} x_{1t} = 0.5x_{2t} + \varepsilon_{1t}, \\ x_{2t} = x_{2,t-1} + \varepsilon_{2t}, \end{cases} \quad (7.10)$$

其中,  $\varepsilon_{it} \sim N(0,1), i=1,2$ , 且  $\varepsilon_{it} (i=1,2)$  相互独立. 试分析序列  $\{x_{1t}\}, \{x_{2t}\}$  之间是否存在协整关系.





# 协整

解 从关联系统 (7.10) 容易得到:

(1) 序列 $\{x_{1t}\} (i=1,2)$ 都是 1 阶单整序列, 即  $x_{1t} \sim I(1), i=1,2$ .

(2)  $\varepsilon_{1t} = x_{1t} - 0.5x_{2t} \sim I(0)$ , 即序列 $\{x_{it}\} (i=1,2)$ 的线性组合平稳.

下面我们通过绘制时序图来进行分析. 具体命令如下, 运行结果如图 7.4 所示.

```
> set.seed(1)
> e1 <- rnorm(1000)
> e2 <- rnorm(1000)
> y2 <- cumsum(e2)
> y1 <- 0.5*y2+e1
```



# 协整

```
> par(mfrow=c(2,2))  
  
> plot.ts(y1,main="y1~I(1)",xlab="(a)",ylab=" ",col=3)  
  
> plot.ts(y2,main="y2~I(1)",xlab="(b)",ylab=" ",col=4)  
  
> plot.ts(cbind(y1,y2),plot.type="single",main="cointegration",  
+ xlab="(c)",ylab=" ",col=c(3,4))  
  
> plot.ts(y1-0.5*y2,main="y1-0.5*y2~I(0)",xlab="(d)",ylab=" ",  
+ col=2)  
  
> par(mfrow=c(1,1))
```

由图 7.4 (a) 和 (b) 可见, 序列 $\{x_{1t}\}, \{x_{2t}\}$ 都呈现一种非平稳的游走态势. 将图 (a), (b)重叠放在一起, 由图 7.4 (c) 可见, 两序列 $\{x_{1t}\}, \{x_{2t}\}$ 之间具有非常稳定的线性关系, 即它们



# 协整

的变化速度几乎一致. 这种稳定的同变关系, 让我们怀疑它们之间具有一种内在的均衡关系. 最后由图 7.4 (d) 知, 两序列  $\{x_{1t}\}, \{x_{2t}\}$  的线性组合的时序图呈现平稳序列关系, 因而序列  $\{x_{1t}\}, \{x_{2t}\}$  具有协整关系.

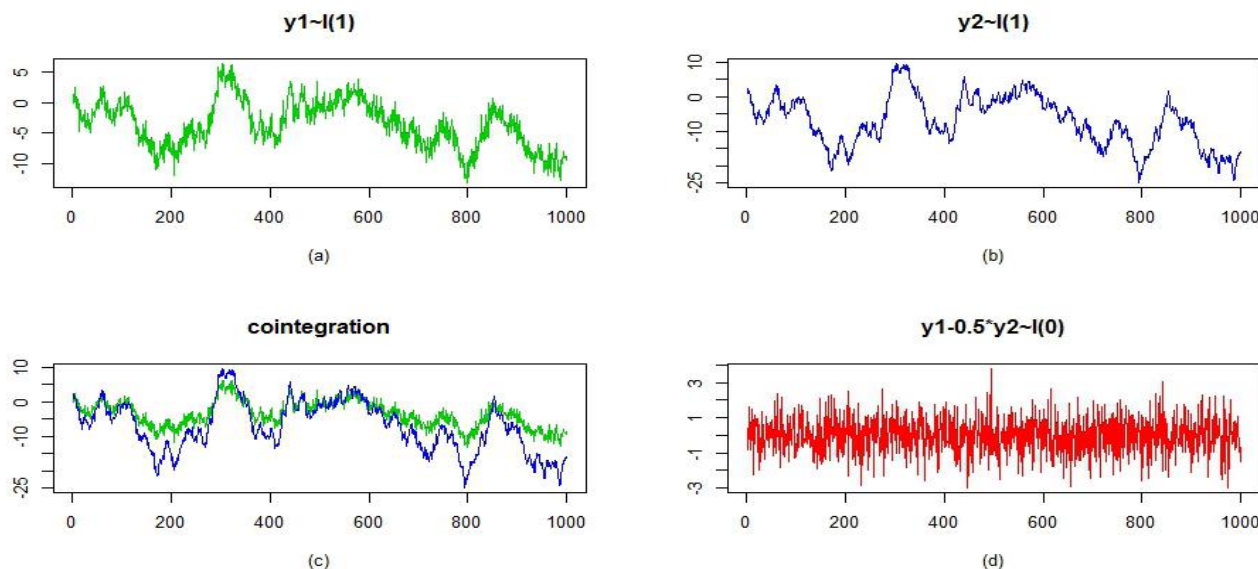


图 7.4 协整关系模拟图

# 协整检验

在 1987 年, Engle 和 Granger 给出了下面协整的严格定义:一般地, 如果向量  $x_t = (x_{1t}, x_{2t}, \dots, x_{nt})^T$  的所有分量序列都是  $d$  阶单整序列, 且存在一个非零向量  $\alpha = (\alpha_1, \alpha_2 \dots \alpha_n)^T$ , 使得  $\alpha^T x_t = \alpha_1 x_{1t} + \alpha_2 x_{2t} + \dots + \alpha_n x_{nt}$  是  $d - b$  阶单整序列, 其中  $0 < b \leq d$ , 那么称向量  $x_t$  的各分量之间是  $(d, b)$  阶协整的, 记为  $x_t \sim CI(d, b)$ . 向量  $\alpha$  称为协整向量.

从协整的定义来看, 协整向量并不唯一. 显然, 如果  $x_t$  有两个协整向量  $\alpha_1$  和  $\alpha_2$ , 那么它们的线性组合  $k_1 \alpha_1 + k_2 \alpha_2$  ( $k_1$  和  $k_2$  不同时为零) 也是  $x_t$  的一个协整向量.

## ■ 协整检验

由协整的定义可以看出, 多个非平稳序列之间能否建立动态



# 协整检验

回归模型的关键是，它们之间是否存在协整关系，因此，对多个非平稳序列建模必须先进行协整检验. 协整检验也称为 Engle-Granger 检验，简称 EG 检验或 EG 两步检验.

由于实际生活中，大多数序列之间不具有协整关系，所以 EG 检验的假设条件可以确定为：

原假设  $H_0$ ：非平稳序列之间不存在协整关系.

备择假设  $H_1$ ：非平稳序列之间存在协整关系.

一般来讲，协整关系主要是通过考察回归残差的平稳性确定的，因而上述假设条件等价于：

原假设  $H_0$ ：回归残差序列  $\{\varepsilon_t\}$  非平稳  $\longleftrightarrow$  备择假设  $H_1$ ：回归残差序列  $\{\varepsilon_t\}$  平稳.



# 协整检验

EG 两步检验的检验步骤如下:

(1) 确定各序列的单整阶数, 并建立回归模型:

$$y_t = \hat{\alpha}_0 + \hat{\alpha}_1 x_{1t} + \hat{\alpha}_2 x_{2t} + \cdots + \hat{\alpha}_n x_{nt} + \varepsilon_t,$$

式中,  $\hat{\alpha}_k$  ( $0 \leq k \leq n$ ) 是普通最小二乘估计值.

(2) 对回归残差序列  $\{\varepsilon_t\}$  进行平稳性检验.

通常对回归残差平稳性的检验采用单位根检验法, 所以假设条件等价于:

原假设  $H_0: \varepsilon_t \sim I(k), k \geq 1 \leftrightarrow$  备择假设  $H_1: \varepsilon_t \sim I(0)$ .

利用 ADF 检验或者 PP 检验的协整分析方法来判断残差序列是否平稳, 如果残差序列是平稳的, 那么回归方程的设定是合理的, 即回归方程的被解释变量和解释变量之间存在



# 协整检验

的长期均衡关系；反之，说明回归方程的被解释变量和解释变量之间不存在稳定的长期均衡关系，即使参数估计的结果很理想，这样的回归也是没有意义的，因为模型本身的设定出现了问题。这样的回归必然是一个伪回归。

例 7.7 分析 1978 年至 2002 年我国农村居民家庭平均每人纯收入的对数序列  $\{\ln x_t\}$  和现金消费支出的对数序列  $\{\ln y_t\}$ ，并进行 EG 检验。

解 读入数据，并绘制时序图。具体命令如下，运行结果见图 7.5。

```
> xz <- read.csv("E:/DATA/CHAP7/7.7.csv",header=T)
```

```
> xt <- ts(log(xz$SR),start=1978)
```



# 协整检验

```
> yt <- ts(log(xz$XF),start=1978)
> plot.ts(cbind(xt,yt),plot.type="single",main=" ",xlab=" ",
+ ylab=" ",lwd=c(1,2))
```

从图 7.5 可以看出,我国农村居民家庭平均每人纯收入的对数序列 $\{\ln x_t\}$ 和现金消费支出的对数序列 $\{\ln y_t\}$ ,具有很强的同变关系,可能二者具有协整关系,下面进行 EG 检验.为此,需确定两序列的单整阶数.首先,对序列 $\{\ln x_t\}$ 和 $\{\ln y_t\}$ ,分别作三种类型的 DF 单位根检验.具体命令及运行结果如下:

```
> adfTest(xt,lag=1,type="nc")
```

Title:

Augmented Dickey-Fuller Test





# 协整检验

Test Results:

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: 1.077

P VALUE:

0.9191

```
> adfTest(xt,lag=1,type="c")
```

Title:

Augmented Dickey-Fuller Test



# 协整检验

Test Results:

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: -1.1822

P VALUE:

0.6125

```
> adfTest(xt,lag=1,type="ct")
```

Title:

Augmented Dickey-Fuller Test



# 协整检验

Test Results:

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: -3.1553

P VALUE:

0.1323

```
> adfTest(yt,lag=1,type="nc")
```

Title:

Augmented Dickey-Fuller Test



# 协整检验

Test Results:

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: 1.3212

P VALUE:

0.9489

```
> adfTest(yt,lag=1,type="c")
```

Title:

Augmented Dickey-Fuller Test

:



# 协整检验

Test Results:

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: -1.4821

P VALUE:

0.5064

```
> adfTest(yt,lag=1,type="ct")
```

Title:

Augmented Dickey-Fuller Test



# 协整检验

Test Results:

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: -3.0817

P VALUE:

0.1603



图 7.5 纯收入的对数序列和现金消费支出的对数序列的时序图

# 协整检验

检验结果表明, 在显著性水平取 0.05 时, 可以认为我国农村居民家庭平均每人纯收入的对数序列和家庭平均每人现金消费支出的对数序列均为非平稳时间序列. 这与图 7.5 显示出来的性质完全一致. 其次, 分别对两序列作 1 阶差分, 再次作单位根检验. 具体命令及运行结果如下:

```
> dx <- diff(xt)
```

```
> for(i in 1:3)print(adfTest(dx,lag=i,type="c"))
```

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:



# 协整检验

Lag Order: 1

STATISTIC:

Dickey-Fuller: -2.23

P VALUE:

0.2416

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 2





# 协整检验

STATISTIC:

Dickey-Fuller: -3.1176

P VALUE:

0.04109

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 3

STATISTIC:



# 协整检验

Dickey-Fuller: -1.8689

P VALUE:

0.3694

```
> dy <- diff(yt)
```

```
> for(i in 1:3)print(adfTest(dy,lag=i,type="c"))
```

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 1



# 协整检验

STATISTIC:

Dickey-Fuller: -3.211

P VALUE:

0.03401

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 2

STATISTIC:



# 协整检验

Dickey-Fuller: -3.1489

P VALUE:

0.03872

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 3

STATISTIC:

Dickey-Fuller: -2.6076



# 协整检验

P VALUE:

0.1079

检验结果表明, 两序列的 1 阶差分序列都是有漂移项的平稳序列, 因此这两个序列都是 1 阶单整序列. 下面构造回归模型. 具体命令及运行结果如下:

```
> moni <- lm(yt~xt)
```

```
> moni
```

Call:

```
lm(formula = yt ~ xt)
```

Coefficients:



# 协整检验

(Intercept) xt

-1.308 1.099

估计农村居民家庭平均每人现金消费支出的对数关于家庭平均每人纯收入的对数的回归方程, 估计结果为

$$\ln y_t = -1.308 + 1.099 \ln x_t + \varepsilon_t. \quad (7.11)$$

最后, 检验残差序列 $\{\varepsilon_t\}$ 的平稳性. 具体命令及运行结果如下:

```
>for(iin 1:3)print(adfTest(moni$residuals,lag=i,type="nc"))
```

Title:

Augmented Dickey-Fuller Test

Test Results:



# 协整检验

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: -2.4715

P VALUE:

0.01707

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:



# 协整检验

PARAMETER:

Lag Order: 1

STATISTIC:

Dickey-Fuller: -2.4715

P VALUE:

0.01707

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:





# 协整检验

Lag Order: 2

STATISTIC:

Dickey-Fuller: -2.0348

P VALUE:

0.04316

Title:

Augmented Dickey-Fuller Test

Test Results:

PARAMETER:

Lag Order: 3



# 协整检验

STATISTIC:

Dickey-Fuller: -1.4538

P VALUE:

0.1464

检验结果表明, 残差序列为无漂移项的 1 阶自相关平稳序列, ADF 检验的  $p$  值为 0.01707.

综上所述, 尽管我国农村居民家庭平均每人纯收入的对数序列和家庭平均每人现金消费支出的对数序列均为非平稳时间序列, 但是它们之间存在协整关系, 所以可建立 (7.11) 式所示的回归模型拟合它们之间的长期均衡关系.



# 本章结构

1. 伪回归

2. 单位根检验

3. 协整

4. 误差修正模型

# 误差修正模型

## ■ 误差修正模型

当序列之间存在协整关系时,说明它们之间存在长期的均衡关系.形象地讲,这些序列之间似乎受到某种作用,使它们协调一致做同变运动.但是就短期而言,序列之间经常会受到某些随机干扰的冲击,可能造成不协调而存在偏差,不过这种偏差会在以后某时期得到修正.为了解释序列之间的短期波动关系和长期均衡关系的内在机理, Hendry 和 Anderson 1977 年提出了误差修正模型 (error correction model), 简称为 ECM 模型. 本小节将简要地介绍 ECM 模型的构造方法. 设非平稳序列  $\{x_{1t}\}$  和  $\{x_{2t}\}$  之间具有协整关系, 即存在非零常数  $\alpha$ , 使得



# 误差修正模型

$$\varepsilon_t = x_{2t} - \alpha x_{1t} \sim I(0). \quad (7.12)$$

将 (7.12) 式写成

$$x_{2t} = \alpha x_{1t} + \varepsilon_t \quad (7.13)$$

进一步由 (7.13) 式得

$$x_{2t} - x_{2,t-1} = \alpha x_{1t} - x_{2,t-1} + \varepsilon_t = \alpha x_{1t} - \alpha x_{1,t-1} - \varepsilon_{t-1} + \varepsilon_t,$$

也即 
$$\nabla x_{2t} = \alpha \nabla x_{1t} - \varepsilon_{t-1} + \varepsilon_t. \quad (7.14)$$

设  $\alpha$  的最小二乘估计值为  $\hat{\alpha}$ , 则  $\hat{\varepsilon}_{t-1} = x_{2,t-1} - \hat{\alpha} x_{1,t-1}$  为上一期误差, 记作  $\text{ECM}_{t-1}$ , 则 (7.14) 式可以写成如下形式:

$$\nabla x_{2t} = \alpha \nabla x_{1t} - \text{ECM}_{t-1} + \varepsilon_t. \quad (7.15)$$

(7.15) 式表明, 序列  $\{x_{2t}\}$  的当期波动值主要受到三个短期



# 误差修正模型

波动的影响: 当期波动值  $\nabla x_{1t}$ 、上一期误差  $ECM_{t-1}$ , 和当期纯随机波动  $\varepsilon_t$ . 为了测量上述三方面的影响, 构造如下 ECM 模型:

$$\nabla x_{2t} = \alpha_0 \nabla x_{1t} - \alpha_1 ECM_{t-1} + \varepsilon_t,$$

其中,  $\alpha_1$  称为误差修正系数, 表示误差修正项对当期波动的修正力度. 根据 (7.15) 式的推导, 可见  $\alpha_1$  应当小于零. 此时, 当  $ECM_{t-1} > 0$ , 也即  $x_{2,t-1} > \hat{\alpha} x_{1,t-1}$  时, 上期真实值大于估计值, 因而导致下期适当减少, 即  $\nabla x_{2t} < 0$ ; 反之, 当  $ECM_{t-1} < 0$ , 也即  $x_{2,t-1} < \hat{\alpha} x_{1,t-1}$  时, 上期真实值小于估计值, 因而导致下期适当增加, 即  $\nabla x_{2t} > 0$ .

例 7.8 续例 7.7, 对 1978 年至 2002 年我国农村居民家庭平均每人纯收入的对数序列  $\{\ln x_t\}$  和现金消费支出的对数



# 误差修正模型

序列  $\{\ln y_t\}$  构造 ECM 模型.

解 首先提取残差, 然后估计模型参数. 具体命令及运行结果如下:

```
> yt.fit <- lm(yt~xt)
> ECM <- yt.fit$residuals[1:24]
> xiuz.fit <- lm(diff(yt)~0+diff(xt)+ECM)
> summary(xiuz.fit)
```

Call:

```
lm(formula = diff(yt) ~ 0 + diff(xt) + ECM)
```

Residuals:



# 误差修正模型

Min 1Q Median 3Q Max

-0.095627 -0.040492 -0.004121 0.059199 0.086703

Coefficients:

Estimate Std. Error t value Pr(> |t|)

diff(xt) 1.12606 0.08192 13.746 2.81e-12 \*\*\*

ECM -0.30140 0.11933 -2.526 0.0193 \*

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0565 on 22 degrees of freedom

Multiple R-squared: 0.8986, Adjusted R-squared: 0.8894





# 误差修正模型

F-statistic: 97.47 on 2 and 22 DF, p-value: 1.166e-11

从上面的估计, 我们得到误差修正模型:

$$\nabla x_{2t} = 1.12606 \nabla x_{1t} - 0.3014 \text{ECM}_{t-1}.$$

从上述误差修正模型来看, 收入的当期波动对消费支出当期波动影响较大, 每增加 1 单位的对数收入, 会增加 1.12606 单位的对数消费支出. 上期误差  $\text{ECM}_{t-1}$  对消费支出影响较小, 单位调整比例为 -0.30140, 而且系数显著性检验显示该系数并不显著非零.

