

Applying Time Series

# 应用时间序列 分析实验

ANALYSIS TO EXPERIMENT

尹剑 白仲林 主编

南开大学出版社

# 应用时间序列分析实验

尹 剑 白仲林 主编

南开大学出版社

天 津

图书在版编目(CIP)数据

应用时间序列分析实验 / 尹剑, 白仲林主编. —天津: 南开大学出版社, 2015.6  
ISBN 978-7-310-04835-9



版权所有 侵权必究

南开大学出版社出版发行

出版人: 孙克强

地址: 天津市南开区卫津路 94 号 邮政编码: 300071

营销部电话: (022) 23508339 23500755

营销部传真: (022) 23508542 邮购部电话: (022) 23502200

\*

天津午阳印刷有限公司印刷

全国各地新华书店经销

\*

2015 年 6 月第 1 版 2015 年 6 月第 1 次印刷

230×155 毫米 16 开本 9.125 印张 2 插页 104 千字

定价: 26.00 元

如遇图书印装质量问题, 请与本社营销部联系调换, 电话: (022) 23507125

# 前 言

时间序列分析作为一种动态数据处理的统计方法，近 50 年来，其理论和实践都有了迅猛发展，深受学术和实务界的褒奖。例如，格兰杰和恩格尔（Granger and Engle, 1986）建立的多元时间序列协整理论等获得了 2003 年的诺贝尔经济学奖。近年来，国内外出版了许多关于时间序列分析的书籍。

为了使初学者掌握和运用时间序列分析的基本方法，熟悉借助 R 语言建立时间序列模型的基本技能，我们编著了这本大学本科生学习时间序列分析的入门实验教材，尤其，本实验教材适用于统计学、应用经济学或工商管理等专业本科生的时间序列分析实践课程。因此，本教材的实践案例涵盖了应用时间序列分析的基本理论内容，选取了社会实践活动中丰富的案例，并在案例分析过程中尽量详尽清楚地表述各种分析方法的步骤和技巧，以便于学生理解、课后练习和类比应用。

本实验教材的编写以研究问题为出发点，将时间序列分析的理论与实践很好地结合起来。为了使内容上更具有实用性，所选择的实验内容和方法系统规范，既有各种专门方法的单向实验，也有各种方法互相比对的设计性实验，加深了学生对理论方法的理解和应用。时间序列分析过程或建立时间序列模型需要借助于统计软件来实现，现有的很多软件都可以用于时间序列分析，但大多是商业软件，购买和使用都会产生较多费用。目前国际上流行使用的 R 语言作为一种开放式交互软件，它完全是免费的。与传统的统计分析软件相比，R 语言是一个开放的统计编程环境，

有很多的共享分析软件包。使用软件 R 不仅能使学生巩固理论知识,而且还可以弥补学生使用其他窗口操作统计分析软件(如 EViews 和 SPSS 等)存在的“黑匣子”缺陷,为时间序列分析实践课程进一步发展奠定技术基础。

本书的主要内容共有九个实验:实验一 R 语言简介,主要介绍 R 软件的基本操作以及时间序列分析中可能使用到的主要函数和扩展包。实验二确定性时间序列模型,以实际的几个案例数据介绍确定性时间序列分析过程。实验三至实验六分别介绍关于平稳性时间序列数据的建模步骤,案例数据依然主要选用实验二中的数据分析结果进行继续研究,使得教材中的专项案例分析具有一定的“连续”性。实验七时间序列波动性分析,介绍了可能存在条件异方差的数据如何进行进一步的建模分析。实验八是单变量时间序列建模综合实验,在以上专项实验的基础上,选取两个数据,进行综合建模实验。实验九是多变量时间序列建模实验,介绍了多变量时间序列建模过程。其中,实验一至实验八由尹剑同志撰写,白仲林同志负责了实验九编写和全书的体系结构安排。

最后,特别感谢天津财经大学统计学系的领导和老师为本教材编写提出了许多宝贵意见,也感谢天津财经大学统计学系的博士生汪玲玲同学为本教材的排版和校对给予了大量帮助。

由于编者水平有限,书稿中还有不足或错误之处,请读者批评指正。

编 者

2015 年 2 月

于天津财经大学振财里

# 目 录

|      |                          |     |
|------|--------------------------|-----|
| 实验一  | R 语言简介 .....             | 1   |
| 实验二  | 确定性时间序列模型 .....          | 13  |
| 实验三  | 平稳时间序列模型的预处理 .....       | 32  |
| 实验四  | 平稳线性 ARMA 模型的识别与定阶 ..... | 44  |
| 实验五  | 平稳线性 ARMA 模型的统计推断 .....  | 53  |
| 实验六  | 时间序列平稳性检验实验 .....        | 71  |
| 实验七  | 时间序列波动性分析实验 .....        | 93  |
| 实验八  | 单变量时间序列建模综合实验 .....      | 105 |
| 实验九  | 多变量时间序列建模实验 .....        | 118 |
| 参考文献 | .....                    | 137 |

# 实验一 R 语言简介

**实验目的：**通过本实验，帮助学生了解软件 R 并熟悉 R 的操作界面；熟悉并掌握软件 R 数据集的不同创建形式；熟悉并掌握利用软件 R 对时间序列数据集进行变换等处理方法。

## 一、软件 R 及其操作界面

### 1.1.1 软件 R 的主要特点和功能

软件 R 作为一个计划 (project)，最早 (1995 年) 是由奥克兰 (Auckland) 大学统计系的简特曼 (Robert Gentleman) 和罗斯·艾赫卡 (Ross Ihaka) 开始编制，目前由 R 核心开发小组 (R Development Core Team) 维护，他们完全自愿、工作努力负责，并将全球优秀的统计应用软件打包提供给大家。软件 R 用的是 S 语言，其运算模式和 C 语言，Basic，MATLAB，Gauss 等很类似。软件 R 是完全免费的自由软件，它的代码公开，可以修改；可以在运行于 UNIX，Windows 和 Macintosh 的操作系统上；R 嵌入了一个非常实用的帮助系统，所有函数都有详细说明，包括变元的

性质、缺省值是什么、输出值是什么、方法的大概说明以及参考文献和作者地址；R 具有很强的作图能力，是集统计分析 with 图形直观显示于一体的软件；R 中包含了大量反映各种新方法的软件包（package），这些软件包可以方便地从 R 主页上下载并安装。这些功能都使得我们在学习和使用软件 R 时非常方便。

### 1.1.2 软件 R 的下载、启动和退出

请登录 R 网站 <http://www.r-project.org>，根据说明从你所选择的镜像站点 CRAN（Comprehensive R Archive Network）来下载适于自己电脑操作系统的 R 的基本程序并安装。

安装完成后，启动 R 软件，界面如图 1.1。我们看到 R 图形用户界面（Graphic Users Interface, GUI）的主窗口，它由三部分组成：主菜单、工具条和 R 的运行窗口。我们的主要工作是在运行窗口通过发布命令来完成的，包括数据集的建立、数据的分析和作图等。

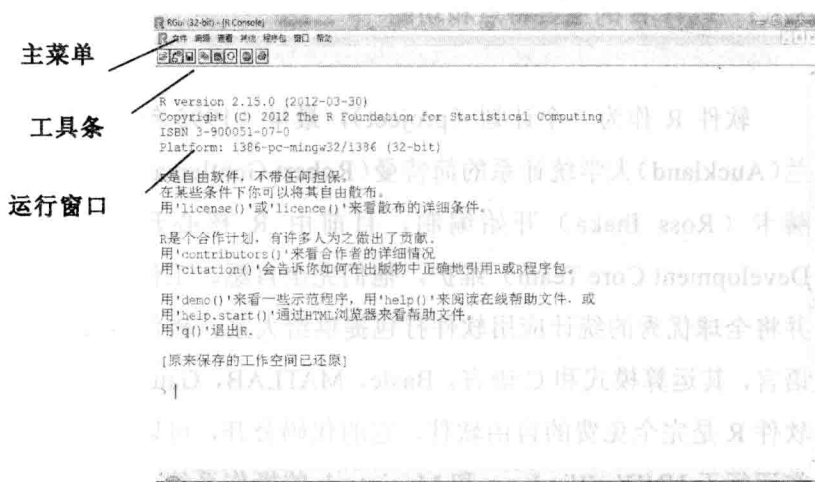


图 1.1 R 软件启动后的主界面窗口



在运行窗口，可以看到一些帮助提示，如：

|                           |                                      |
|---------------------------|--------------------------------------|
| <code>help()</code>       | 得到相应函数的帮助，例如 <code>help(plot)</code> |
| <code>help.start()</code> | 格式的关于 R 的帮助文件                        |
| <code>demo()</code>       | 得到 R 提供的几个示例                         |
| <code>q()</code>          | 退出 R                                 |

### 1.1.3 软件 R 的操作界面

R 的基本界面是一个交互式命令窗口，命令提示符是一个大于号“>”，在提示符后输入指令，结果马上显示在指令下面。R 的指令主要有两种形式：表达式和赋值运算（用“<-”或“=”表示）。在命令提示符后键入一个表达式表示计算此表达式并显示结果。赋值运算把赋值号右边的值计算出来赋给左边的变量。R 是区分大小写的，所以 `x` 和 `X` 是不同的变量名称。

下面用一些例子来看 R 软件的特点。假设现在已经进入了 R 的交互式窗口。如果没有打开的图形窗口，在 R 中，用 `> x11()` 可以打开一个作图窗口。然后，输入以下语句：

```
> x1= 0:100;x2= x1*2*pi/100;y = sin(x2);  
> plot(x2,y,type= "l")
```

则在图形显示窗口将给出如图 1.2 的图形显示。这些语句可以绘制正弦曲线图。其中，“=”是赋值运算符。`0:100` 表示一个从 0 到 100 的等差数列向量。第二个语句可以看出，我们可以对向量直接进行四则运算，计算得到的 `x2` 是向量 `x1` 的所有元素乘以常数  $2\pi/100$  的结果。从第三个语句可看到函数可以以向量来输入，并可以输出一个向量，结果向量 `y` 的每一个分量是自变量 `x2` 的每一个分量的正弦函数值。从最后一个语句可以看出函数的调用也很自由，可以按位置给出自变量，也可以用“自变量名=”的形

式指定自变量值，这样可以使用缺省值。

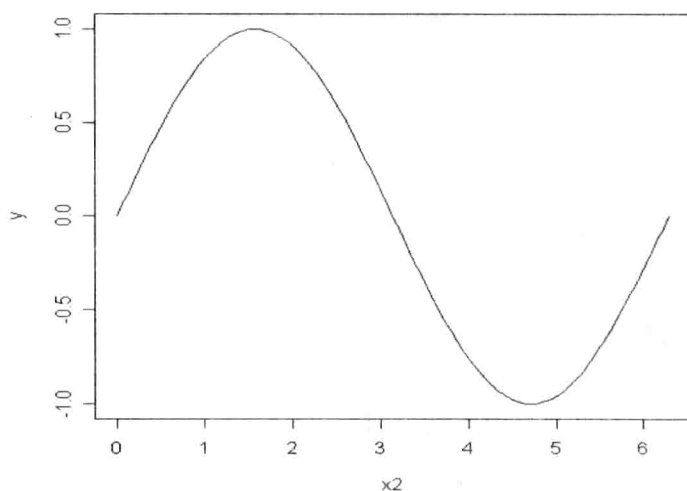


图 1.2 正弦曲线图

## 二、软件 R 中数据集的创建

### 1.2.1 利用 data.frame 函数创建数据集

数据框是 R 中类似 SAS 数据集的一种数据结构。它通常是矩阵形式的数据，但矩阵各列可以是不同类型的。数据框每列是一个变量，每行是一个样品的观测值。

数据框可以用 `data.frame()` 函数生成，各自变量变成数据框的成分，自变量可以命名，成为变量名。例如：

```
> d=data.frame( name=c("赵", "钱", "孙", "李", "王")),
```

```
age=c(20,21,22,21,20), height=c(170,171,175,165,181),
gender=c("男", "女", "男", "女", "男")
```

```
> d
```

结果显示如下：

|   | name | age | height | gender |
|---|------|-----|--------|--------|
| 1 | 赵    | 20  | 170    | 男      |
| 2 | 钱    | 21  | 171    | 女      |
| 3 | 孙    | 22  | 175    | 男      |
| 4 | 李    | 21  | 165    | 女      |
| 5 | 王    | 20  | 181    | 男      |

引用数据框元素的方法与引用矩阵元素的方法相同，可以使用下标或下标向量，也可以使用名字或名字向量。如 `d[1:2, 2:3]`。数据框的各变量也可以用按列表引用（即用双括号`[[ ]` 或`$`符号引用）。

数据框的变量名由属性 `names` 定义，此属性一定是非空的。数据框的各行也可以定义名字，可以用 `rownames` 属性定义。如：

```
> names(d)
```

结果显示如下：

```
[1] "name" "age" "height" "gender"
```

结果显示如下：

```
> rownames(d)
```

```
[1] "1" "2" "3" "4" "5"
```

### 1.2.2 导入文件中的数据集

(1) 读取数值型变量。为了从外部文件读入一个数值型向量，R 提供了 `scan()` 函数。如果指定了 `file` 参数（也是第一参数），则

从指定文件读入，缺省情况下读入一个数值向量，文件中各数据以空白分隔，读到文件尾为止。例如：

```
> cat(1:12, "\n", file="E:/work/result.txt")
> x = scan("E:/work/result.txt")
```

如果文件中是一个用空白分隔的矩阵（或数组），我们可以先用 `scan()` 把它读入到一个向量，然后用 `matrix()` 函数（或 `array()` 函数）转换。如：

```
> y = matrix(scan("E:/work/result.txt"), ncol=3, byrow=T)
```

（2）读取数据框。如果要读取一个数据框，R 提供了一个 `read.table()` 函数。它只要给出一个文件名，就可以把文件中用空白分隔的表格数据每行读入为数据框的一行。比如，文件 `E:\d.txt` 中内容如下：

```
Zhou      15      3
Li Ming    9      李明
Zhang     10.2    Wang
```

用 `read.table` 读入：

```
> x1 = read.table("E:/d.txt", as.is=T)
> x1
```

结果显示如下：

```
      V1      V2      V3
1 Zhou    15      3
2 Li Ming  9      李明
3 Zhang   10.2    Wang
```

则读入结果为数据框。

函数可以自动识别表列是数值型还是字符型，并在缺省情况下把字符型数据转换为因子（加上 `as.is=T` 可以保留字符型不转换）。

函数自动为数据框变量指定“V1”“V2”这样的变量名，指定“1”“2”这样的行名。可以用 `col.names` 参数指定一个字符型向量作为数据框的变量名，用 `row.names` 参数指定一个字符型向量作为数据框的行名。

如果要读入的文件中首行表示的是变量名称，而不是观测到的数据值，则此时读取数据时需要在命令 `read.table` 中加入参数 `header` 值。比如，文件 `E:\d1.txt` 中内容如下：

|     | 长度  | 高度  | 次数 |
|-----|-----|-----|----|
|     | 2.5 | 650 | 16 |
|     | 3.8 | 520 | 23 |
| ... |     |     |    |

用 `read.table` 读入：

```
> x2=read.table("E:/d1.txt", header=T)
```

```
> x2
```

结果显示如下：

|     | 长度  | 高度  | 次数 |
|-----|-----|-----|----|
| 1   | 2.5 | 650 | 16 |
| 2   | 3.8 | 520 | 23 |
| ... |     |     |    |

(3) 读取 `excel` 数据。读取 `excel` 数据时，可以先用 `excel` 将数据打开，另存为 `csv` 格式的数据，然后使用 `read.csv` 命令来导入到 R 中。一般情况下，使用默认的参数就可以。具体的参数可以通过在 R 命令窗口中输入 `?read.csv` 来查询。

比如将 `c:\file.csv` 导入到 R 中，可以使用：

```
>x3=read.csv("c:/file.csv")
```

如果文件中有缺失数据，软件 R 以 `NA` 来表示。

### 1.2.3 导出 R 中的数据

对于文件读取和写入的工作，R 语言使用工作目录来完成，可以使用 `getwd()`（获得工作目录）命令来找到工作目录。使用 `getwd()` 命令也可以改变工作目录，例如 `getwd("D:/data")` 将工作目录改变为“D:/data”文件夹。也可以手动的改变：点击菜单中的“文件”，然后选择“改变工作目录选项”即可。

对于读取的文件，如果不在工作目录下，可以指定文件的目录，也可以把文件移动到 R 的工作目录下，直接进行操作。

最常见的工作是把一个矩阵或数据框以数字的矩形网格方式写入文件中，而且还可能保留行列的标签。这可以通过函数 `write.table` 和 `write` 来完成。函数 `write` 仅可以写出一个矩阵或向量的特定列（和对一个矩阵进行转置）。函数 `write.table` 更为便利，它可把一个数据框（或一个可以强制转换为数据框的对象）以包含行列标签的方式写出。而如果想把 R 中的数据框导出到指定文件夹下的 excel 文件，可以使用函数 `write.csv`。更具体的参数使用方法可通过 `?write.table` 和 `?write.csv` 来查询函数的帮助文件。

例如 R 中有一个名称为 `df1` 的数据框，如果想将它导出到文件夹“D:\documents”中，并保存为文件名为 `dataframe1` 的文本文件，则可以通过命令：

```
>write.table(df1, "D:/documents/dataframe1.txt")
```

也可以通过命令：

```
>write.csv(df1, "D:/documents/dataframe1.csv")
```

将数据框 `df1` 保存为一个 excel 格式的文件。

## 三、时间序列数据集的相关处理方法

### 1.3.1 时间序列的输入

软件 R 中，使用函数 `ts` 来创建时间序列对象，用函数 `as.ts` 将一个对象转换为时间序列，用函数 `is.ts` 判断一个对象是否为一个时间序列对象。

函数 `ts` 的调用格式为：

```
ts(data = NA, start = 1, end = numeric(0), frequency = 1)
```

其中参数 `data` 取值为时间序列数据值，可以是向量或矩阵或数组的形式；`start` 和 `end` 分别是时间序列起点和终点时间；参数 `frequency` 指明时间序列的单位时间数据个数。如：

```
> x=1:10
```

```
> x
```

结果显示如下：

```
[1] 1 2 3 4 5 6 7 8 9 10
```

输入以下命令：

```
> ts(x)
```

结果显示如下：

```
Time Series:
```

```
Start = 1
```

```
End = 10
```

```
Frequency = 1
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

输入以下命令：

```
> par(mfrow=c(1,2))
```

```
> plot(x)
```

```
> plot.ts(x)
```

结果如下图 1.3 所示。

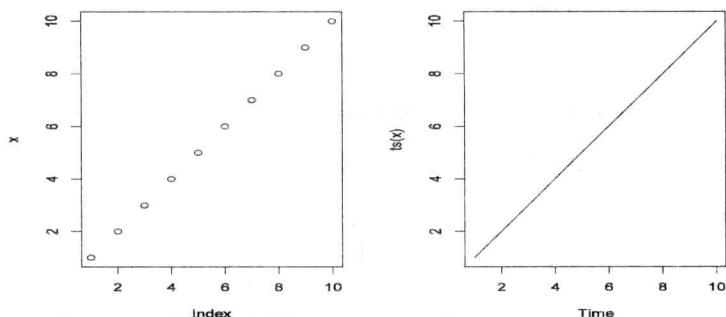


图 1.3 plot 命令绘制的图形比较：向量对象（左图）和时间序列对象（右图）

从图 1.3 可以看出，不同的对象应用 plot 绘图命令得到的图形是不同的，转换为时间序列对象的 ts(x) 序列的下标也被转换为时间指标，并且图形默认为折线图而不是散点图。

R 中常用的绘图命令 plot 函数主要绘制二维散点图，plot.ts(x) 函数绘制 x 的时间序列曲线图，x 可以是多元的，但要求序列必须有相同的频率和时间。ts.plot(x) 函数功能同 plot.ts，但如果 x 是多元的，序列可以有不同的频率和时间。如果数据对象已经转换为时间序列对象，则绘图命令 plot 和 plot.ts 的功能相同，否则 plot 绘制的是散点图而不是折线图。

图 1.3 左侧的图形是 plot(x) 命令绘制的散点图，右边是 plot.ts(x) 命令绘制的时间序列折线图。plot.ts 函数将向量转换为时间序列对象再进行绘图，因而绘制的不再是散点图，而是将向量的下标转换为时间，将数据转换为时间序列数据绘制时间序列曲



线图。

R 中的 `plot` 函数的功能很强大，它可以绘制多种类型的图形，例如绘制区间  $[-2\pi, 2\pi]$  上的函数  $y=\sin(x)+\cos(x)$  图像，则可以通过 `plot` 函数轻松实现。先应用函数 `function` 构建给出的函数关系，再通过 `plot` 函数来绘制图像（如图 1.4 所示），具体的命令如下：

```
>y=function(x){y=sin(x)+cos(x);y}
```

```
>plot(y,-2*pi,2*pi)
```

结果如下图 1.4 所示。

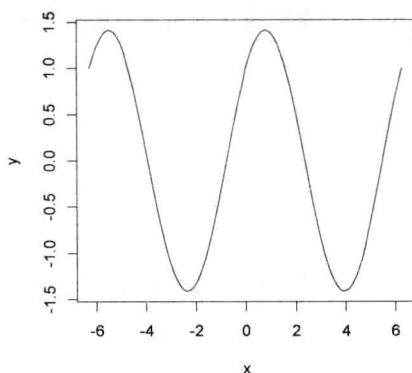


图 1.4 函数曲线图

### 1.3.2 对已有的数据集中数据的处理

如果从外部文件读入一个数据集，也可以先转化为我们所需要的时间序列对象，如：

```
> x=read.table("D:/2.txt",header=T) #从 D 盘提取文件
```

```
> x= ts(x,start=c(1990,1),end=c(2005,12),frequency=12)
```

```
> plot.ts(x)
```

结果显示如下图 1.5 所示：