

第 1 章 引言

1.1 时间序列的特点

本书研究的时间序列主要是根据时间先后对同样的对象按等间隔时间收集的数据,比如每日的平均气温、每天的销售额、每月的降水量、每秒钟容器的压力等.虽然这些序列所描述的对象本身(比如容器压力)可能是连续的,但由于观测值并不连续,这种时间序列数据称为离散(discrete)时间序列.当然也有连续(continuous)时间序列,它是不间断地取值得到的时间序列,比如地震波和一些状态控制(比如月球车是否苏醒用 1-0 表示的状态)的连续观测值,但对象不一定是连续的(比如 0-1 状态).和用于简单最小二乘回归的横截面数据不同,时间序列的观测值有可能是相关的,比如今天的物价和昨天的物价相关,本月的 CPI 和上个月的 CPI 相关等.

根据待研究的时间序列变量个数的不同,时间序列分为一元时间序列和多元时间序列.在使用时间序列进行预测时往往使用一个(一元)或一组(多元)时间序列的历史数据来预测其未来的值,这在一元时间序列的应用中尤其明显.在多元时间序列中也有用一些时间序列作为输入来预测另一些时间序列的情况.这些特点使得时间序列更注重序列本身的自相关,并且利用过去时间序列的模式来预测未来.因此,研究时间序列本身的性质就很重要了.对于没有规律的时间序列,或者受到其他未知的或不可观测变量严重影响的时间序列,人们基本上无能为力.而且,为了有足够的关于序列模式的信息,时间序列分析通常需要较多的历史数据.只有那些比较“规范”而且信息量较大的时间序列才有可能被建模并且用于预测.此外,一些时间序列模型有很强的数学假定,但这些根据经验的假定永远无法用数据验证.如果不注意到这些局限性,则会导致滥用和误导.

什么是较“规范”的序列呢?比方说,对于一元时间序列来说,就是在进行一系列差分变换之后可以转变为“平稳”的序列,“平稳”是时间序列分析中的一个重要概念.一般地,平稳的序列可以用可操作的数学模型来描述和处理.而对于多元时间序列,则

希望各个序列的线性组合是平稳的, 这样人们可以用数学方式来建模. 这里所用的数学模型通常都是线性模型, 或者是部分线性模型.

一般来说, 自然界中具有规律的降水、气压、气温、湿度等序列是比较规范的时间序列. 在与人类社会活动相关的序列中, 食品和日用品的销售等是比较有规律的, 某些宏观经济数据也是如此. 但诸如股票、期货等贸易及金融数据就很难有较明确的规律性, 这些时序的变动依赖于许多其他因素, 而且与产生数据的市场的成熟程度有关.

下面介绍一些时间序列的例子.

1.2 时间序列例子

例 1.1 美国年度经济数据 (sf.csv). 该数据来自 Fisher(1994)^①. 该数据的变量为 DATE (相应于年末)、PRICEIND(1900—1938 年道琼斯工业及交通运输指数, 拼接到 1939—1985 年的纽约证交所综合指数)、PREMIUM(每年美国股票溢价)、USVEL2(纽约证交所成交率) 和 GROWTH(美国工业生产增长指数).

图1.1为例1.1序列的点图. 该图是用下面语句生成的:

```
w=read.csv("sf.csv")
w=ts(w[, -1], start=c(1899,1), freq=1)
plot(w, main="", type="o", pch=16)
```

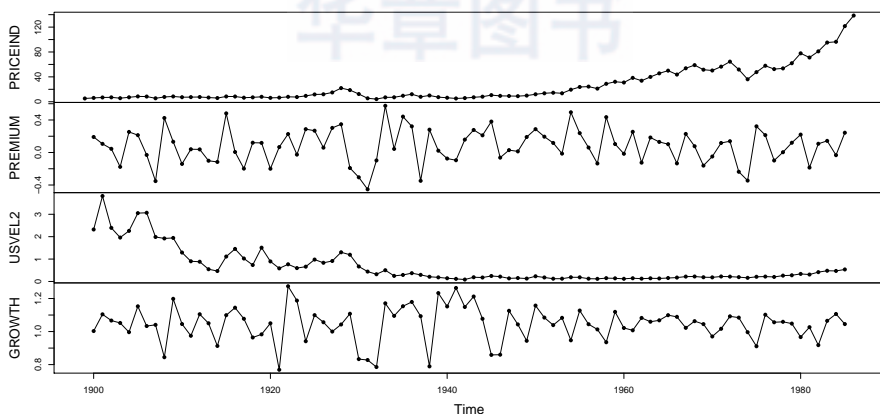


图 1.1 例1.1美国年度经济数据

^① 可以从网页 <http://qed.econ.queensu.ca/jae/1994-v9.S/fisher/data.sf> 下载.

这是一组经济时间序列, Fisher (1994) 研究了这些序列之间的关系. PRICEIND 和 USVEL2 序列随着时间变化表现出上升和下降的长期趋势. PREMIUM 和 GROWTH 序列的波动较大, 且呈现一定的周期性. 在宏观经济分析领域, 这些时间序列数据扮演了很重要的角色. 通过对这些序列的预测和分析可以更好地为国家经济的管理服务. 本书的很多方法主要是用于经济金融领域的时间序列.

例 1.2 2000 年 1 月到 2012 年 10 月新西兰 5 个城市 (Auckland, Christchurch, Dunedin, Hamilton, Wellington) 的月度总降水数据 (NZRainfall.csv). 该数据可从网上下载^①.

图1.2为例1.2序列的点图. 该图是用下面语句生成的:

```
w=read.csv("NZRainfall.csv")
x=ts(w[,-1],start=c(2000,1),freq=12)
plot(x,ylab="Monthly rainfall (mm)",type="o",pch=16,nc=1,main="")
title("New Zealand Rainfall")
```

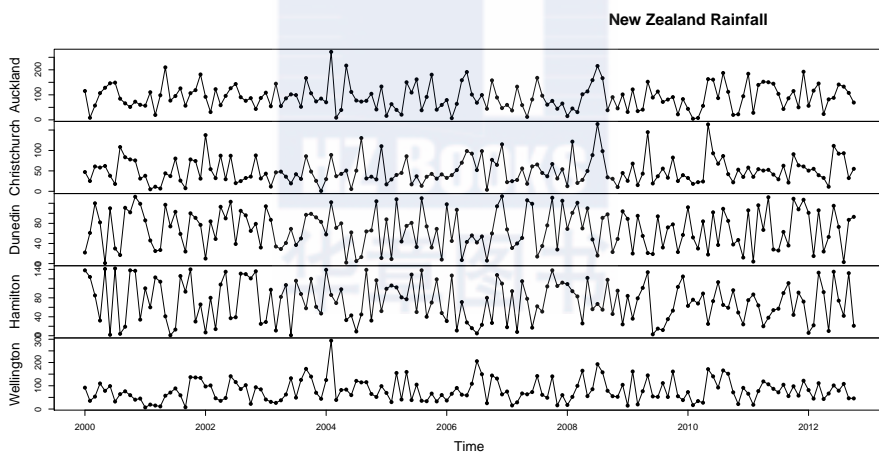


图 1.2 例1.2新西兰 5 个城市 2000 年 1 月到 2012 年 10 月的月度总降水量

这是一组降水数据, 与例1.1比较, 这组数据有比较强的周期性, 可以看出降水量随月度变化呈现出较有规律的波动. 从平均降水量上看, Dunedin 和 Hamilton 的月度降水量比较相似, 主要集中在 0mm 及 120mm 之间. Auckland 和 Wellington 降水量的峰值相对较高. 总体来说, 月度降水量属于比较规律的自然现象, 所以这几个序列并没有

^① <http://new.censusatschool.org.nz/resource/time-series-data-sets-2013/>.

明显的总体上升或下降趋势变化. 诸如降水和气温等气象数据是时间序列中比较有规律的一族.

例 1.3 这是世界上 15 个城市的月度最高和最低温度 (2000-01 ~ 2012-10). 这里我们仅显示 Bogota 和 HongKong 两个城市的月度最低温度 (TempWorld2.csv).^① 从图1.3可以看出这两个时间序列有相同的周期, 但又有很不相同的形状细节, 下面是读入数据和点图1.3的代码.

```
u=read.csv("TempWorld2.csv")[,c(5,11)]
ut=ts(u,start=c(2000,1),frequency=12)
plot(ut)
```

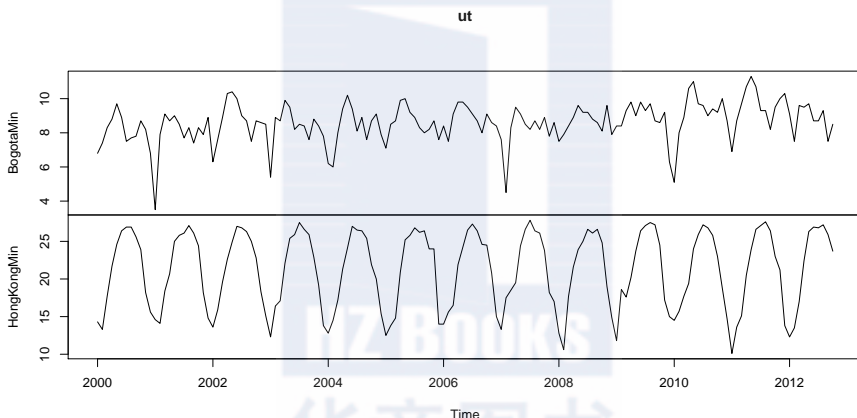


图 1.3 例1.3中 Bogota 和 HongKong 两个城市的月度最低温度

例 1.4 这是美国联邦政府的数据, 关于有效的联邦基金利率 (Effective Federal Funds Rate)(1954-07-01 ~ 2017-02-01)^② 和 3 个月国库券二级市场利率 (Treasury Bill: Secondary Market Rate, TB3MS) (1943-01-01 ~ 2017-02-01).^③

从图1.4可以看出这两个时间序列没有什么明显的周期, 但非常相似, 下面是读入数据和点图1.4的代码.

```
v1=read.csv("FEDFUNDS.csv")
```

^① <http://new.censusatschool.org.nz/resource/time-series-data-sets-2013/>.

^② <https://fred.stlouisfed.org/series/FEDFUNDS/downloaddata?cid=118>.

^③ <http://research.stlouisfed.org/fred2/series/TB3MS/>.

```
v1t=ts(v1[,-1],start=c(1954,7),frequency=12)
v2=read.csv("TB3MS.csv")
v2t=ts(v2[,-1],start=c(1934,1),frequency=12)
par(mfrow=c(1,2))
plot(v1t)
plot(v2t)
```

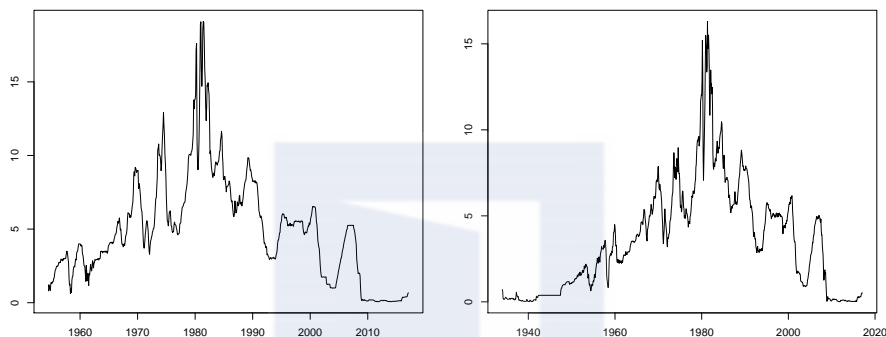


图 1.4 例1.4中 Funds Rate 和 TB3MS 两个序列图

前面的几个例子都是关于离散时间序列的, 下面的描述性例子是关于连续时间序列的. 这种连续时间序列不是本书研究的目标. 此外, 在本节的时间序列点图中, 凡是离散时间序列的点图都包含了观测值点及它们的连线, 以后的时间序列图一般都不标出观测值的点, 但一定标出观测点之间的连线 (因此看上去是连续的).

例 1.5 状态为 0 和 1 的连续监控时间序列. 图1.5为该序列的点图. 该序列是模拟出来的.

这个时间序列的观测对象是离散的 (0-1 过程), 但序列本身是连续的. 本书将不对这种数据进行分析, 引入这个例子仅仅是作为知识的介绍.

1.3 R 软件入门

1.3.1 简介

统计是数据科学, 分析数据必须要用软件, 而使用最方便、统计资源最丰富的开源性软件就是免费的 R 软件.

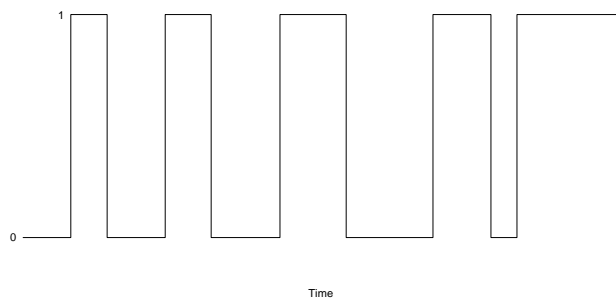


图 1.5 例1.5状态为 0 和 1 的连续监控时间序列

R 软件 (R Development Core Team, 2011) 用的是 S 语言, 其运算模式和 C 语言、Basic、Matlab、Maple、Gauss 等类似. R 软件是免费的自由软件, 它的代码公开, 可以修改, 十分透明和方便. 大量国外新出版的统计方法专著都附带有 R 程序. R 软件有强大的帮助系统, 其子程序称为函数. 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、方法的说明以及参考文献和作者的地址. 大多数函数都有例子, 把这些例子的代码复制并粘贴到 R 界面就可以立即得到结果, 十分便捷. 反映新方法的各种程序包 (package) 可以从 R 网站上下载, 更方便的是联网时通过 R 软件菜单的“程序包 – 安装程序包”选项可直接下载程序包.

软件必须在使用中学, 仅仅从软件手册中学习是不可取的, 正如仅仅用字典和语法书无法学会讲外语一样. 笔者用过众多的编程软件, 没有一个是课堂或者手册学的, 都是在分析数据的实践中学会的. 笔者在见到 R 软件时, 已经接近“耳顺”之年, 但在一天内即基本掌握它, 一周内可以熟练编程和无障碍地实现数据分析目的. 耄耋糊涂之翁尚能学懂, 何况年轻聪明的读者乎! 近年出现了大量关于 R 软件的英文参考资料, 很容易下载, 其代码可以复制和粘贴到 R 软件上 (笔者当年可没有如此幸运). 其中有一个网站上有 Vincent Zoonekynd 编写的《Statistics with R》, 实际上是网络书籍, 也可以下载其 PDF 版本^①, 很容易使用; 还有一本书是《Modern Applied Statistics with S》(Venables and Ripley, 2002), 可以下载其第 4 版的 PDF 版本^②, 该书通过 R 软件来介绍许多现代统计方法, 是非常好的一本书, 有一个 R 自带的程序包 (MASS) 就是以这本书的内容编写的 (MASS 为该书名字的缩写).

^① 网址为: http://zoonek2.free.fr/UNIX/48_R/all.html.

^② 网址为: http://www.planta.cn/forum/files_planta/modern_applied_statistics_with_s_192.pdf.

安装和运行小贴士

- 登录 R 网站 (<http://www.r-project.org/>)^①, 根据说明从你所选择的镜像网站下载并安装 R 的所有基本元素.
- 向左边变量赋值可以用 “=” 或者 “<-”, 还可以用 “->” 向右赋值.
- 运行时可以在提示码 “>” 后逐行输入指令. 如果回车之后出现 “+” 号, 则说明你的语句不完整 (要在 + 号后面继续输入) 或者已输入的语句有错误.
- 每一行可以输入多个语句, 之间用半角分号 “;” 分隔.
- 所有代码中的标点符号都用半角格式 (基本 ASCII 码). R 的代码对于字母的大小写敏感. 变量名字、定性变量的水平以及外部文件路径和名字都可以用中文.
- 不一定非得键入你的程序, 可以粘贴, 也可以打开或新建以 R 为扩展名的文件 (或其他文本文件) 作为运行脚本, 在脚本中可以用 `Ctrl+R` 来执行 (计算) 光标所在行的命令, 或者仅运行光标选中的任何部分.
- 出现的图形可以用 `Ctrl+W` 或 `Ctrl+C` 来复制并粘贴 (前者像素高), 或者通过菜单存成所需的文件格式.
- 如果在运行时按 `Esc` 键则会终止运行.
- 在运行完毕时会被问到 “是否保存工作空间映像?”, 如果选择 “保存”, 下次运行时, 这次的运行结果还会重新载入内存, 不用重复计算, 缺点是占用空间. 如果已经有脚本, 而且运算量不大, 一般都不保存. 如果你点击了 “保存”, 又没有输入文件名, 这些结果会放在所设或默认的工作目录下的名为 `.RData` 的文件中, 你可以随时找到并删除它.
- 注意, 从 PPT、PDF 或 Word 文档之类非文本文件中复制并粘贴到 R 上的代码很可能存在由这些软件自动变换的字体、首字母大写或者左右引号等造成的 R 无法执行的问题.
- R 中有很多常用的数学函数、统计函数以及其他函数. 可以通过在 R 的帮助菜单中选择 “手册 (PDF 文件)”, 在该手册的附录中找到各种常用函数的内容.
- 在 R 界面, 你可以用问号加函数名或数据名得到该函数或数据的细节, 比如用 “?`lm`” 可以得到关于线性模型函数 “`lm`” 的各种细节. 另外, 如果想查看 MASS 程序包中的稳健线性模型 “`rlm`”, 在已经打开该程序包时 (用 `library(MASS)` 打开, 用 `detach(package:MASS)` 关闭), 可用 “?`rlm`” 得到该函数的细节. 如果 MASS 没

^① 网上搜索 “R” 即可得到其网址.

有打开^①, 或者不知道 `rlm` 在哪个程序包, 可以用 “`??rlm`” 得到其位置. 如果对于名字不清楚, 但知道部分字符, 比如 “`lm`”, 可以用 “`apropos("lm")`” 得到所有包含 “`lm`” 字符的函数或数据.

- 如果想知道某个程序包有哪些函数或数据, 可以在 R 的帮助菜单上选择 “Html 帮助”, 再选择 “Packages” 即可得到你的 R 上装载的所有程序包. 这个 “Html 帮助” 很方便, 可以链接到许多帮助 (包括手册等).
- 有一些简化的函数, 如加、减、乘、除、乘方 (“`+`”、“`-`”、“`*`”、“`/`”、“`^`”) 等, 可以用诸如 “`?"+`” 这样的命令得到帮助 (不能用 “`?+`”).
- 可以写关于代码的注释: 任何在 “`#`” 号后面作为注释的代码或文字都不会参与运行.
- 你可能会遇到无法运行过去已经成功运行过的一些代码, 或者得到不同结果的情况. 原因往往是这些程序包经过更新, 一些函数选项 (甚至函数名称和代码) 都已经改变, 这说明 R 软件的更新和成长是很快的. 解决的办法是查看该函数, 或者查看提供有关函数的程序包来探索一下究竟.
- 不同的程序包经常包含有同样名字的函数, 比如, 程序包 `ipred` 及程序包 `adabag` 都有函数 `bagging()`, 虽然在分类问题上它们有类似原理, 但输出及预测等各方面都有区别. 这时如果两个程序包都属于开启状况就可能会出现某些代码运行困难, 必须关闭其中不需要的一个程序包 (如使用代码 `detach(package:adabag)` 关闭程序包 `adabag`) 以执行你需要的程序包的函数.
- 有一个名为 RStudio 可以自由下载的软件能更方便地用几个窗口来展示 R 的执行、运行历史、脚本文件、数据细节等过程.

1.3.2 动手

如果你不愿意弄湿游泳衣, 即使你的老师是世界游泳冠军, 即使你在教室里听了几百小时的课, 你也永远学不会游泳. 软件当然要在使用中学. R 软件的资源丰富, 功能非常强大, 我们不可能也没有必要把每一个细节都弄明白, 有很多功能笔者到现在也没有用到, 或者是因为没有需要, 或者是因为有替代方法. 我们都有小时候读书的经验, 能看懂多少就看懂多少, 很少查字典, 后来长大了, 在开始学外语时, 由于大量单词不会才对不认识的单词查字典. 实际上, 读外语时, 在有一定单词量的情况下, 能猜就不查字典可能是更好的学习方式.

^① 通常为了节省内存以及避免变量名字混杂, 应该在需要时打开相应的程序包, 不需要时关闭.

本书最后有附录“使用 R 软件练习”，提供了一些笔者为练习而编的代码，如果全部一次运行，要不了一分钟。但希望读者在每运行一行之后就进行思考，一般人都能够在两天内完全理解这些代码。如果在学习以后章节的统计内容时不断实践，R 语言就会成为你自己的语言了。

建议初学 R 者，在读本书之前，务必花些时间，运行一下这些代码！

1.4 本书的内容

本书着重于通过现有的数学模型对真实数据进行分析。这就需要针对有各种特点的时间序列引进各种数学模型。人们不可能证明真实的时间序列会满足某个数学模型的数学假定，各种模型中所有关于数据的数学假定仅仅是人们头脑中对真实现象的某种近似，这些近似和实际问题的差距是永远不可能知道的，正如所有科学理论仅是对未知真理的近似一样。因此，对于一个时间序列会有多种模型可以拟合，结果也不尽相同，这是很正常的。此外，因为每个模型都是试图描述时间序列的某个或某些方面，因此，根据不同的目的，模型的选择也会不同。

本书第 2 章介绍一元时间序列的基本概念和 ARIMA 模型，这些概念和模型对于理解后面的章节很重要。但是，如果第 2 章的基本内容已经知道，或者大概了解，就不用在这一章花太多工夫，完全可以等到需要时再来查阅。第 2 章的习题也主要是与熟悉基本概念有关。

第 3 章为通过实际数据来应用第 2 章的知识做数据分析，以得到各种拟合及预测结果。此外第 3 章还介绍了 ARIMA 模型之外的其他模型及相关的拟合和预测等数据分析。第 3 章的习题主要是提供一些实际数据，希望读者创造性地模仿该章的例子来做一元时间序列数据分析。

第 4 章的状态空间和 Kalman 滤波涉及一些计算方法。很多一元时间序列的分析程序都潜在地用到了第 4 章的方法，但不一定明显注明。当然，在第 8 章则会用到多元状态空间模型。因此，第 4 章可以考虑跳过去，等到需要时再来参阅。本书也没有提供关于第 4 章内容的习题。

第 5~7 章主要介绍计量经济学中常用的单位根检验、研究长记忆过程的 ARFIMA 模型及研究时间序列波动的 GARCH 模型。由于这三章主要是描述有关的数学概念和模型本身，因此没有安排习题。第 6 章根据情况可以跳过。第 5 章和第 7 章内容的主要应用体现在第 8 章中。

第 8 章主要介绍多元时间序列的基本概念和模型。第 8 章最后是对实际多元时

间序列数据的拟合和预测, 涉及了前面各章的很多概念, 在习题中提供了许多实际数据让读者熟悉如何处理多元时间序列.

第 9 章是关于非线性时间序列的一些模型.

第 10 章是谱分析简介. 前面九章的内容主要是从时间域角度分析, 方法展开主要基于时序的自相关函数性质和稳定性; 谱分析则从另一个角度来研究时间序列, 我们称之为频率域角度, 频率域分析把时序看成多组正弦曲线的叠加过程. 该章主要是引入一些数学概念, 介绍谱分析的基本思路, 并将时间域和频率域分析的关联性和一致性建立起来. 第 10 章不涉及具体数据的分析, 也没有习题, 可以仅作参考之用.

1.5 习题

1. 从网上下载 R 软件.
2. 在联网的情况下, 打开 R, 点击“帮助-CRAN 主页-Packages”看看今天有多少可供使用的程序包 (package), 再点击按照时间顺序排列的程序包列表, 看看今天又增加了多少新程序包 (包括更新的).
3. 在联网的情况下, 打开 R, 点击“帮助-CRAN 主页-Task Views”看看有多少你感兴趣的领域或方向, 再点击你感兴趣的领域, 看看有多少软件包可用.
4. 在 R 中 (不必要联网), 通过语句 `?lm` 来看线性模型 (linear model) 函数 `lm()` 的用法, 然后看说明后面的例子, 把这些例子的代码逐行粘贴到 R 的运行界面, 查看结果.
5. 在 R 中, 用语句 `mylm=fix(lm)` 或者 `mylm=edit(lm)` 就可以把函数 `lm()` 变成你自己的名为 `mylm()` 的函数, 而且还可以修改.
6. 在联网状态, 在 R 菜单中点击“程序包 - 安装程序包”后, R 会要求你选择镜像网站, 选择完了就会出现一个很长的程序包列表, 你可以选择一个程序包或多个程序包 (按住 Ctrl 键) 来安装, 这种安装会自动安装这些程序包以及它们所需要的支持程序包 (即使你没有选).
7. 你也可以通过键入命令来安装程序包. 你可以试着键入下面的命令安装一个和多个程序包:

```
install.packages("Ecdat"); install.packages(c("TSA", "vars"))
```

这时, R 会要求你选择镜像网站, 选择完了就会自动安装这些程序包以及它们所需要的支持程序包 (即使你没有选).

8. 根据上题在 R 中安装了程序包 `Ecdat`, 用 `library(Ecdat);?Macrodat` 打开程序包 `Ecdat` 所提供的数据 `Macrodat` 的说明, 然后用 `plot(Macrodat)` 语句画出有关的 7 个时间序列的图. 当然, 你也可以用语句

```
plot(Macrodat,plot.type="single")
```

把它们画到一张图上, 看看有什么区别.

9. 把上一题最后一个画图语句 `plot(Macrodat,plot.type="single")` 改变成语句

```
plot(Macrodat,plot.type="single",type="o"),
```

即增加选项 `type="o"`, 看看会产生何种不同的点图. 由此, 根据帮助 `?plot` 来了解相应于 `type` 的不同选项所产生的不同图形模式.

10. 建立工作目录是很方便的, 如果有了工作目录, 存取文件都在这个目录文件夹之下, 不用再指名路径. 有两种方式建立工作路径:

- (1) 敲入诸如 `setwd("D:/mywork")` 设立你自己的工作目录 (当然这个文件夹必须存在).
- (2) 在 R 的菜单中, 点击“文件 – 改变工作目录”后, 会出现路径窗口, 你自己选择适合的工作目录文件夹.

如果你不知道目前的工作目录是什么, 可以用 `getwd()` 来获得.

11. 存取文件. 在选择工作目录, 并且用 `library(Ecdat)` 载入程序包 `Ecdat` 之后, 先点击“文件 – 新建程序脚本”, 会出现一个编辑窗口. 然后用语句

```
write.csv(Macrodat, "Macrodat.csv",row.names=F)
```

把数据 `Macrodat` 存入硬盘的目录中. 你再到该目录下寻找这个文件, 并打开查看, 会发现没有时间存入 (因为原数据就没有时间列). 如果要提取这个文件的数据, 加上数据说明中注明的时间段, 可以用语句

```
w=read.csv("Macrodat.csv")  
w=ts(w,start=c(1959,1),frequency=4)
```

得到. 这时点击 `w` 就可以看到有时间列的数据. 在关闭程序脚本前, 可以把这个脚本存为以“.R”结尾的文件^①.

12. 你可以用命令 `history()` 来查看工作时所用命令的记录, 可以把这个记录存到你的硬盘上, 以后还可以重复原先的工作.

^① 目前, 新建脚本保存的次数有限 (可能是个缺陷), 最好起了文件名之后关闭, 再点击“文件 – 打开程序脚本”重新打开它, 这时就不会有保存的次数问题了.