arXiv:2301.01790v1 [stat.ME] 4 Jan 2023

使用 R 中的 smooth 程序包进行平滑预测

Ivan Svetunkov[a]

[a]营销分析与预测中心、管理科学系、兰卡斯特大学，英国

摘要

R 中有许多与预测相关的程序包，受欢迎程度各不相同，其中最著名的是 forecast，它实现了若干重要的预测方法，例如自回归积分滑动平均模型、指数平滑状态空间模型、TBATS 等。然而，现有功能的主要问题在于，若要出于研究目的修改已实现的模型，灵活性不足。R 程序包 smooth 引入了一种新的单变量预测方法，以单一误差源（SSOE）状态空间形式实现指数平滑和自回归积分滑动平均模型，并提供用于实验和时间序列分析的高级功能。该程序包以 SSOE 模型为基础，通过纳入解释变量、多个频率，并引入高级预测工具对其加以扩展。本文阐述了该程序包背后的理念，并展示其主要函数的工作方式。

关键词：预测、指数平滑、ETS、ARIMA、ADAM、 R

1. 引言

R（R Core Team，2022年）是学术界最受欢迎的编程语言之一，拥有许多实现了各种方法的预测相关软件包。其中较为知名的是 forecast 程序包（Hyndman 和 Khandakar，2008年），它实现了经典统计预测模型，例如 ETS（基于指数平滑背后的单一误差源状态空间框架的误差、趋势、季节性模型；Hyndman 等人，2002年、2008年）、Theta（Assimakopoulos 和 Nikolopoulos,

*通讯地址：I. Svetunkov，兰卡斯特大学管理学院营销分析与预测中心，兰卡斯特，兰开夏郡，LA1 4YX，英国。电子邮箱：i.svetunkov@lancaster.ac.uk (Ivan Svetunkov)

在线发表于 www.openforecast.org 2023年1月6日

2000年）、TBATS（De Livera等，2011年）等。其中一些函数也已在 `fable`程序包（O'Hara-Wild等，2021年）中实现。该程序包还实现了用于自动选择 ARIMA 阶数的 `auto.arima()` 函数。其他实现 ARIMA 的软件包还包括`stats` （R Core Team，2022年）、 `robustarima` （Kaluzny 和 TIBCO Software Inc.，2021年）、 `tfarima` （Gallego，2021年）和 `fable` （O'Hara-Wild等，2021年）。这些软件包都提供了适用于特定情境的即用型函数，并且已被证明效果出色，但它们不具备动态模型领域研究所需的灵活性，也未能为单变量预测模型提供整体性方法。

为了解决这些问题，早在 2016 年，我就开发了一个 `smooth`包，实现了单一误差源框架中的模型，并提供了灵活性，使人们能够在单变量预测模型领域开展高级实验（例如，使用高级损失函数、引入解释变量、更改模型结构等）。本文介绍了 `smooth` 函数背后的主要思想，总结了这些函数的用途，以及如何将其用于预测和分析。

2. 单一误差源框架

`smooth` 函数所基于的主要模型在 Svetunkov（2022）专著中有详细说明。该模型以 Hyndman 等人（2008）模型为基础。这里我们只概述主要思想。首先介绍最常用的纯加性模型，它是 `smooth` 包中大多数函数的基础（Svetunkov，2023）。其形式如下：

$$\begin{aligned} y_t =& \mathbf{w}' \mathbf{v}_{t-\mathbf{l}} + \epsilon_t \\ \mathbf{v}_t =& \mathbf{F} \mathbf{v}_{t-\mathbf{l}} + \mathbf{g} \epsilon_t \end{aligned}, \tag{1}$$

其中，$\mathbf{w}$ 是测量向量，$\mathbf{F}$ 是转移矩阵，$\mathbf{g}$ 是持久性向量，$\mathbf{v}_{t-\mathbf{l}}$ 是滞后分量向量，$\mathbf{l}$ 是滞后向量，用于定义$\mathbf{v}_t$ 各分量在时间上的移位方式。与 Hyndman 等（2008）提出的传统状态空间模型不同， `smooth` 所采用的方法依赖于滞后分量，而非它们的转移。在传统模型中，状态向量$\mathbf{v}_t$ 始终取决于前一个观测值对应的向量值，信号则按照矩阵$\mathbf{F}$从一个分量转移到另一个分量。传统框架和所提出的框架所对应的 ETS 和 ARIMA 模型完全相同，

但我们的方法简化了计算。例如，考虑 ETS(A,A,A)模型，其表达式如下（Hyndman 等（2008））：

$$\begin{aligned} y_t &= l_{t-1} &+ b_{t-1} &+ s_{t-m} &+ \epsilon_t \\ l_t &= l_{t-1} &+ b_{t-1} & &+ \alpha\epsilon_t \\ b_t &= & b_{t-1} & &+ \beta\epsilon_t \\ s_t &= & & s_{t-m} &+ \gamma\epsilon_t \end{aligned}. \tag{2}$$

其中， y_t 是实际值， l_{t-1} 是水平， b_{t-1} 是趋势， s_{t-m} 是周期性为 m 的季节性成分（例如，对于按月统计的数据， m 为 12，表示每 12 个月重复一次）， α、 β 和 γ 是平滑参数， ϵ_t 是独立同分布误差项。根据 (1)，模型 (2) 可写为：

$$\begin{aligned} y_t &= \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} l_{t-1} \\ b_{t-1} \\ s_{t-m} \end{pmatrix} + \epsilon_t \\ \begin{pmatrix} l_t \\ b_t \\ s_t \end{pmatrix} &= \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} l_{t-1} \\ b_{t-1} \\ s_{t-m} \end{pmatrix} + \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} \epsilon_t, \end{aligned} \tag{3}$$

而按照 (Hyndman 等人，2008年) 的形式，则为：

$$\begin{aligned} y_t &= \begin{pmatrix} 1 & 1 & 1 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} l_{t-1} \\ b_{t-1} \\ s_{1,t-1} \\ s_{2,t-1} \\ \vdots \\ s_{m,t-1} \end{pmatrix} + \epsilon_t \\ \begin{pmatrix} l_t \\ b_t \\ s_{1,t} \\ \vdots \\ s_{m,t} \end{pmatrix} &= \begin{pmatrix} 1 & 1 & \mathbf{0}'_{m-1} & 0 \\ 0 & 1 & \mathbf{0}'_{m-1} & 0 \\ 0 & 0 & \mathbf{0}'_{m-1} & 1 \\ \mathbf{0}_{m-1} & \mathbf{0}_{m-1} & \mathbf{I}_{m-1} & \mathbf{0}_{m-1} \end{pmatrix} \begin{pmatrix} l_{t-1} \\ b_{t-1} \\ s_{1,t-1} \\ \vdots \\ s_{m,t-1} \end{pmatrix} + \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ 0 \\ \vdots \\ 0 \end{pmatrix} \epsilon_t. \end{aligned} \tag{4}$$

由于式（4）中的矩阵规模以及模型的递归性质，当季节性数据的 m 高于 24 时，应用该模型会带来较大的计算开销

这是因为需要进行大型矩阵乘法。当需要包含多个季节性成分的模型时，问题会更加严重（例如参见 Taylor，2003年），因为这会引入多个季节性指数，使式（4）中的矩阵规模进一步增大。式（1）解决了这一问题，因为引入额外成分只会使维度按所添加成分的数量成比例增加。请注意，对传统状态空间模型的任何扩展都会增加其维度，不可避免地导致计算困难。因此，所提出的模型（1）更具实用性和灵活性，也因此被用于开发 `smooth` 函数。

此外，还有一种更通用的状态空间形式，不仅涵盖纯加法模型，还涵盖纯乘法模型和混合模型。Hyndman 等（2008）在第 4 章以及 Svetunkov（2022）在第 7.1 节对此进行了讨论。本文不讨论这种模型，只指出 `smooth` 中实现的乘法型和混合型 ETS 模型是基于该形式的。

此外，Snyder（1985年）指出，ARIMA 模型也可以写成 SSOE 状态空间形式。Hyndman 等（2008）随后在第 11 章对此进行了更详细的讨论，之后 Svetunkov 和 Boylan（2020）利用这一形式在供应链场景中实现并应用了 ARIMA（通过 `ssarima()` 和 `auto.ssarima()` 函数，来自 `smooth` 包）。在此基础上，Svetunkov（2022）以状态空间形式（1）实现了 ARIMA（第 9 章）。因此，所提出的框架并不局限于 ETS 模型，还可以扩展，例如纳入 ETS+ARIMA 的组合。

拼图的最后一块是回归模型。它也可以表示为 SSOE 状态空间形式 (1)，例如 Koehler 等人（2012）就讨论过这一点。Hyndman 等人（2008）第 9 章和 Svetunkov（2022）第 10 章对此进行了讨论。

这一切意味着，状态空间模型 (1) 为使用指数平滑状态空间模型、自回归积分滑动平均模型、回归及其任意组合提供了一个统一的框架。这通过 `smooth` package 的 `adam()` function 实现（Svetunkov，2023），支持以下功能：

1. 指数平滑状态空间模型；2. 自回归积分滑动平均模型；3. 回归；4. 时变参数回归；5. (1)、(2) 与 (3) 或 (4) 的组合；

6. 自动选择/组合指数平滑状态空间模型的状态；7. 自动选择自回归积分滑动平均模型的阶数；8. 回归的变量选择；9. 误差项的正态与非正态分布；10. 自动选择最合适的分布；11. 多重季节性；12. 模型中的发生部分，用于处理数据中的零值（适用于间歇性需求）；13. 对分布尺度进行建模（GARCH 类模型，例如参见 Engle，1982年）；14. 处理参数估计值的不确定性；15. 使用上述任一元素进行预测。

所有这些主题都在 Svetunkov（2022）中有所涉及，因此这里我们不会重点介绍`adam` function。不过，模型 (1) 有一些特殊情况，在 `smooth` package 中实现了特定功能。本文将讨论其中最重要的几种。

3. 时间序列分解

虽然 `stats` package 已经实现了经典分解函数，但我创建了一个新函数，用来处理具有多个季节周期的数据。它名为 `msdecompose()`。它与经典分解（Warren M. Persons，1919年）的逻辑完全相同，但可用于具有多个季节周期的数据。用户可以通过设置参数 `lags`来定义数据包含哪些周期，并通过参数 `type`选择季节性类型。例如，可以按以下方式将 `msdecompose()` 应用于来自 `forecast`package 的半小时电力需求数据 `taylor`：

```
taylorDecomp <- msdecompose(taylor, lags=c(48,336),
                            type="m")
```

这将生成一个可用于进一步分析的对象。根据该对象绘图会生成多张图（详情请参阅`plot.smooth()` 的文档），其中最有意思的一张——时间序列成分图——见图 1。

经典分解通常无法产生清晰的成分——图 1 中的残差表明存在季节性，因为

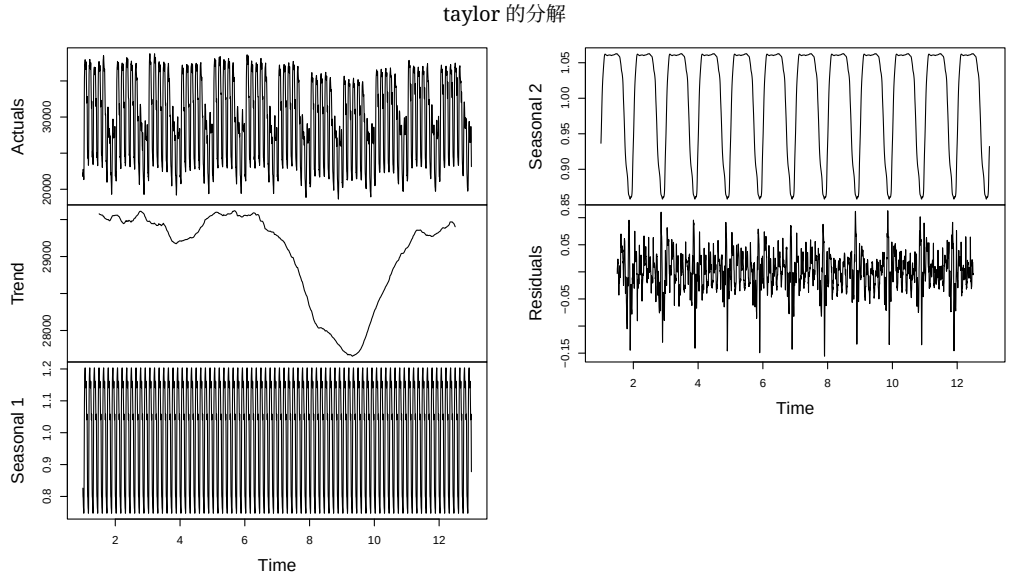


图 1：根据 msdecompose() 函数对多重季节性时间序列进行分解。

该方法假设季节性成分是恒定的。尽管如此，它仍可作为时间序列分析的起点。在 `smooth`中，它用于季节性数据情况下 ETS 的状态初始化，并且主要用于处理多重季节性数据。不过，如果研究人员对使用季节分解进行预测感兴趣，该函数会生成 `smooth` 类的对象，该对象支持 `forecast()` 方法：先为分解后的数据中的趋势成分生成预测值，然后根据估计的季节性成分重构序列。

4. 指数平滑

指数平滑是需求计划中最常用的预测方法之一（Weller 和 Crone, 2012）。如前所述，ETS 是所有指数平滑方法的基础，并被视为预测领域的学术标准（多年来，它一直应用于所有主要的预测竞赛，包括 Makridakis 和 Hibon，2000年；Athanasopoulos 等人，2011年；Makridakis 等人，2020年、2022年）。Hyndman 等（2008）提出的传统 ETS 模型假设误差项

服从正态分布。它由 ets() 包中的函数forecast 和 ETS() 包中的函数 fable实现。 smooth包中的对应函数称为 es()，但它基于状态空间模型（1），而非传统模型。此外， ets() 仅支持 15 种 ETS 模型，而 es() 则实现了理论上所有可能的 30 种 ETS 模型。该函数还支持对模型参数进行精细调整，允许通过变量 persistence设置平滑参数值，通过 initial 设置初始值，通过 initialSeason 设置季节性指数，并通过参数 B 预先指定优化所用的参数值。此外，该函数还支持通过参数 xreg 纳入解释变量，类似于 stats 包中的 arima 函数；用户可以通过 initialX 调整回归变量的系数，并使用 Sagaert 和 Svetunkov（2022）算法，根据 ETS 模型残差并通过参数 regressors ，按照信息准则选择最合适的回归变量。

就 ETS 组件的选择而言， es() 默认使用的机制可概括为以下步骤：

1. 对数据应用 ETS(A,N,N)，计算信息准则（IC）；2. 对数据应用 ETS(A,N,A)，计算 IC。如果其值低于（1）中的结果，则说明数据中存在某种季节性成分，进入步骤（3）。否则，转到（4）；3. 应用 ETS(M,N,M) 模型并计算 IC。如果其值低于前一步的结果，则说明数据呈现乘法季节性。转到（4）；4. 使用前面步骤选定的加法趋势组件和季节性成分拟合模型，其中二者可以是“N”“A”或“M”。计算新模型的 IC，并将其与目前为止的最佳 IC 进行比较。如果其值低于之前应用模型的准则值，则说明数据中存在趋势成分。否则，不需要趋势成分；5. 根据步骤（1）至（4）构建模型集合，应用这些模型，并选择 IC 最低的模型。

这种组件选择方法称为分支定界，因为它不会遍历所有可能的模型，而是考虑模型分支。例如，如果不存在季节性，就可以将相应的组件设为“N”，从而排除季节性模型分支，并减少

待测试的模型数量，从30个减少到仅10个（包括在四个步骤中已测试的模型）。

与其他任何 smooth 函数类似， es() 支持多种方法，包括用于模型可视化诊断的 plot() 和用于预测的 forecast() 。为演示它们的工作方式，我们将 es() 应用于 AirPassengers包中的数据。

```
AirPassengersETS <- es(AirPassengers, h=12, holdout=TRUE)
```

在上面的代码中，我们指定了 12 步的预测范围，并要求在模型训练中排除最后 12 个观测值，从而创建一个测试集（留出集），以查看模型在这部分数据上的表现。我们可以对模型进行诊断，看看是否存在任何明显的问题可以解决：

```
par(mfcol=c(2,2))
plot(AirPassengersETS, c(1,2,4,6))
```

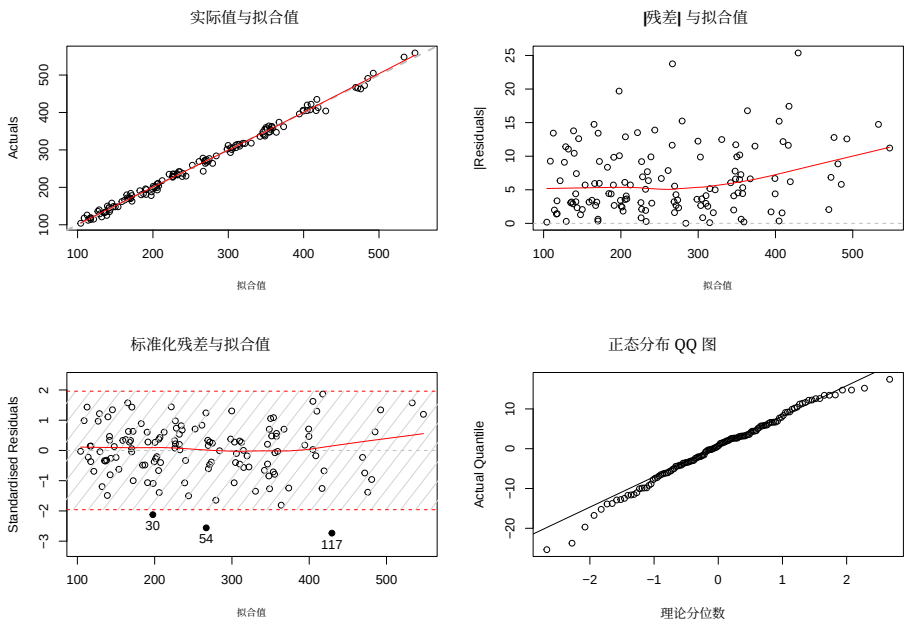


图2：es()函数在 AirPassengers 数据上自动选择的 ETS(A,M,M) 模型的诊断图。

结果图如图2所示。本文并不旨在解决模型存在的问题，只是演示如何使用 `smooth` 函数。通过这些图，可以分析残差中可能存在的问题，例如异方差性、自相关、离群值、模型设定错误等。可以通过纳入解释变量和/或更改模型中使用的变换来解决这些问题。解决潜在问题后，研究者可以根据估计模型生成预测值，这通过 `generics` 包中的 `forecast()` 方法实现（Wickham 等，2022年）。但与针对 `ets()` 和 `ETS()`的预测不同，`smooth` 中的方法支持多种选项，用户可以从多种预测区间中进行选择（参见 `forecast.smooth()` 方法的文档），还可以生成单侧区间（这在低容量数据上的纯乘法模型中很有用，因为其下界通常等于零），并生成累积预测（这在库存管理中的安全库存计算中很有用）。我们将使用参数的默认值来生成参数预测区间：

```
plot(forecast(AirPassengersETS, h=12))
```

上述代码将生成图3所示的图。

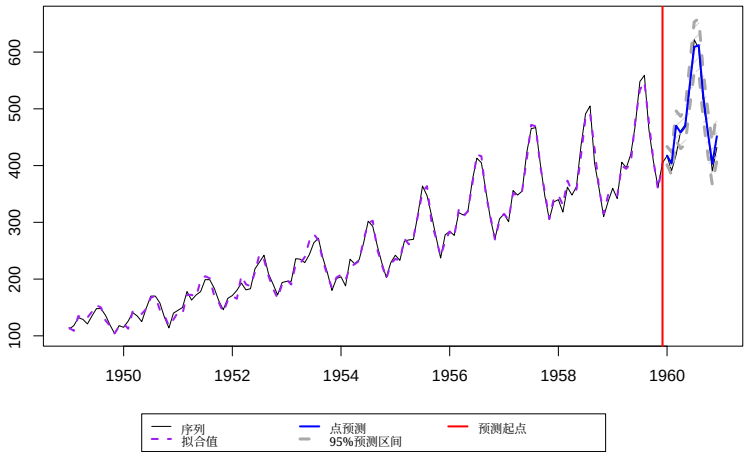


图3：es()函数对AirPassengers数据生成的forecast。

图 3 展示了所选模型对数据的拟合效果、生成的点预测（留出集部分的粗体蓝色实线）以及生成的预测区间（留出集中的灰色区域）。

延续指数平滑这一主题， smooth 还通过 ces() 函数实现了 Svetunkov 等（2022）提出的复指数平滑，其功能与 es() 类似，并支持相同的一组方法。

最后，如前所述， adam() 也实现了 ETS 模型，并支持更多功能。 adam() 中的默认 ETS 与 es() 的主要区别在于，前者支持正态分布以外的其他分布，并且在乘法误差模型中默认使用伽马分布。

5. 自回归积分滑动平均模型

另一个在预测中经常使用的重要模型是 ARIMA（Box 和 Jenkins，1976年）。 smooth 包中有多个函数可用于实现 SSOE 状态空间形式的 ARIMA。

ssarima()（状态空间 ARIMA）函数实现了 Hyndman 等（2008）第 11 章中讨论的形式的状态空间 ARIMA。用于对状态空间 ARIMA 进行阶数选择的函数名为auto.ssarima()。它不依赖任何统计检验，而是根据信息准则选择阶数。Svetunkov 和 Boylan（2020）对该模型和选择机制均作了说明。

msarima() (多重季节性 ARIMA) 函数基于状态空间模型 (1)，引入滞后分量，从而大幅缩小转移矩阵的规模。这使得我们能够将大型多重季节性 ARIMA 模型应用于数据。需要注意的是，因此该模型的转移矩阵、观测向量和状态向量的构成方式与 Hyndman 等（2008）中的不同。一般情况下，它们如下所示（Svetunkov，2022年，第 9 章）：

$$\begin{aligned} \mathbf{F} = \begin{pmatrix} \eta_1 & \eta_1 & \dots & \eta_1 \\ \eta_2 & \eta_2 & \dots & \eta_2 \\ \vdots & \vdots & \ddots & \vdots \\ \eta_K & \eta_K & \dots & \eta_K \end{pmatrix}, \mathbf{w} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \\ \mathbf{g} = \begin{pmatrix} \eta_1 + \theta_1 \\ \eta_2 + \theta_2 \\ \vdots \\ \eta_K + \theta_K \end{pmatrix}, \mathbf{v}_t = \begin{pmatrix} v_{1,t} \\ v_{2,t} \\ \vdots \\ v_{K,t} \end{pmatrix}, \quad \mathbf{l} = \begin{pmatrix} 1 \\ 2 \\ \vdots \\ K \end{pmatrix}, \end{aligned} \tag{5}$$

其中， η_j 是模型 ARI 部分的 j^{th} 多项式， θ_j 是 j^{th} MA 参数， K 是 ARI/MA 多项式中阶数最高者的阶数。为了更好地理解该模型的构造方式，请看 ARIMA(1,1,2) 的例子，它可以写作：

$$(1-\phi_1 B)(1-B)y_t = (1+\theta_1 B+\theta_2 B^2)\epsilon_t, \tag{6}$$

其中 B 是后移算子。该模型可以写成状态空间形式（推导过程参见 Svetunkov，2022 的第 9 章）：

$$\begin{aligned} y_t &= v_{1,t-1} + v_{2,t-2} + \epsilon_t \\ v_{1,t} &= (1+\phi_1)(v_{1,t-1} + v_{2,t-2}) + (1+\phi_1+\theta_1)\epsilon_t. \\ v_{2,t} &= -\phi_1(v_{j,t-j} + v_{2,t-2}) + (-\phi_1+\theta_2)\epsilon_t \end{aligned} \tag{7}$$

为了说明模型 (7) 可以表示为形式 (1)，我们需要设置以下矩阵和向量：

$$\begin{aligned} \mathbf{F} = \begin{pmatrix} 1+\phi_1 & 1+\phi_1 \\ -\phi_1 & -\phi_1 \end{pmatrix}, \mathbf{w} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \\ \mathbf{g} = \begin{pmatrix} 1+\phi_1+\theta_1 \\ -\phi_1+\theta_2 \end{pmatrix}, \mathbf{v}_t = \begin{pmatrix} v_{1,t} \\ v_{2,t} \end{pmatrix}, \quad \mathbf{l} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}. \end{aligned} \tag{8}$$

最后，如前所述， `adam()`函数也支持 ARIMA[1]，形式与 `msarima()` 相同。这三个函数的 ARIMA 语法相似：用户需要通过 `lags`向量定义模型的季节滞后阶数，列出模型应包含的所有季节频率；还需要通过 `orders` 变量定义模型阶数，该变量通常接受形如 `orders=list(ar=c(1,2,3),i=c(1,2,3),ma=c(1,2,3))`的命名列表，用于定义模型在相应滞后阶数下的自回归、差分和移动平均部分的阶数。ARIMA 阶数的设计方式使研究人员能够引入任意数量的滞后阶数，例如支持双重和三重季节性 ARIMA。请注意，由于其公式设计， `ssarima()` 无法处理高频数据，而且随着季节滞后阶数 m的增加，运行速度会变慢。

以下是一个用户自定义的 SARIMA(0,2,2)(0,2,2)$_{12}$ 模型应用于相同 `AirPassengers` 数据的示例：

[1]请注意，要在 adam() 中关闭模型的指数平滑状态空间模型部分，用户需要指定 model="NNN"。

```
AirPassengersARIMA <- msarima(AirPassengers, lags=c(1,12),
orders=list(i=c(2,2),ma=c(2,2)), h=12,  holdout=TRUE)
```

为了查看模型对数据的拟合效果，我们可以使用 `plot`函数，并指定 `which=7`:

```
plot(AirPassengersARIMA,7)
```

之后，我们将得到图4所示的图表。

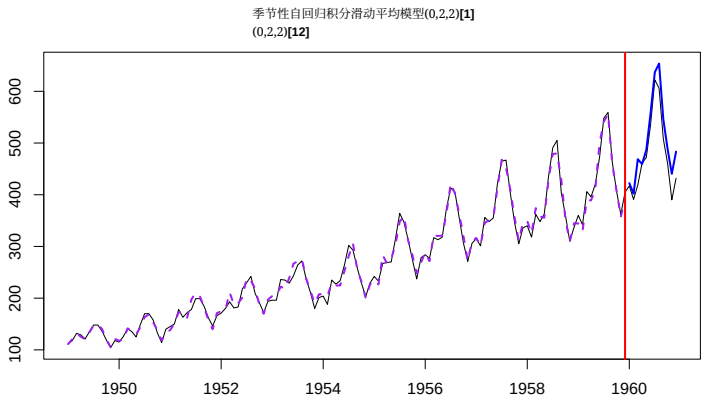


图4：由 msarima() 函数生成的 AirPassengers数据预测结果。

此外，所有 `smooth` 函数都支持以下三种初始化机制之一：

1. 优化——在优化阶段估计状态向量的初始值；2. 反向预测——将参数优化后的模型应用于反转后的数据，从最后一个观测值开始递归至第一个观测值，从而生成初始值；3. 手动设置——由用户提供初始值。

这些由函数中的 `initial` 参数控制。对于 ARIMA，由于任务较为复杂，`initial="backcasting"` 通常比另外两种方法运行得更快、效率更高。

如果研究者需要一个自动选择阶数的 ARIMA 模型，可以使用 auto.ssarima()、 auto.msarima()，该方法将按照 Svetunkov 和 Boylan（2020）以及 Svetunkov（2022）第 15.2 节中所述的流程，选择使指定信息准则最小化的模型。对于 adam()，可通过在 orders 参数的列表中添加变量 select=TRUE 来启用自动选择机制。

使用上述三个函数生成的 ARIMA 模型支持其他 smooth 函数所提供的所有方法，包括 plot()、 actuals()、fitted()、 residuals()和 forecast()。

6. 模拟函数

smooth 包支持的另一类重要函数是模拟函数。它们可以根据假设模型生成数据。该包提供了几个函数：

1. sim.es() 可根据所选 ETS 模型生成数据，并设置 persistence、initial 和 initialSeason参数；2. sim.ces() 可根据复指数平滑数据生成过程生成数据；3. sim.ssarima() 可根据 ARIMA 模型生成数据，并允许设置模型阶数、自回归参数、移动平均参数以及常数项的值（具体为截距还是漂移项，取决于差分阶数）。

如果未指定参数，则会随机选取。上述所有函数均支持多种误差项分布，也支持使用手动创建的分布。下面举例说明如何使用后者：

```
customFunction <- function(n, mu, sd){
return(log(abs(rnorm(n, mu, sd))));}
x <- sim.es("ANN", obs=100,
randomizer="customFunction", mu=0, sd=1)
```

模拟函数允许生成所需数量的序列，具体数量由 nsim 参数控制。

最后，该软件包还实现了 simulate() 方法，该方法从已经估计的模型中提取参数，以便根据该模型生成模拟数据。为了了解其工作原理，我们根据在第 4 节中估计的AirPassengersETS 模型生成数据：

```
x <- simulate(AirPassengersETS, obs=120, nsim=5)
plot(x)
```

上述代码将生成五个时间序列，其中每个序列看起来都与图5所示的序列相似。

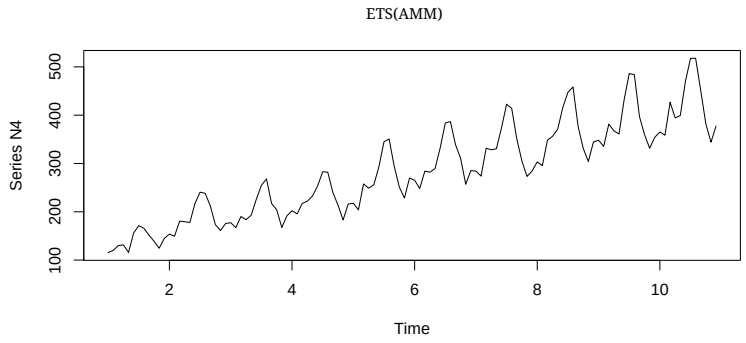


图5：AirPassengersETS 模型生成的模拟数据。

从图5中的图可以看出，生成的时间序列表现出与原始时间序列相似的行为。它甚至具有相似的季节性形状，但趋势有所不同，增长速度比原始数据慢。

7. 其他函数

该软件包还实现了其他几个函数，但它们不在本文的讨论范围内。不过，其中有两个值得一提。

简单移动平均（SMA）函数 `sma()` 已在状态空间模型 (1) 中实现。该函数基于 Svetunkov 和 Petropoulos（2018）的论文；他们指出，SMA(p) 存在一个底层的 AR(p) 过程，其参数限制为 $\phi_j = \frac{1}{p}$，适用于所有 $j = 1, \dots, p$。该函数还支持通过信息准则自动进行阶数选择，相关讨论见原论文。

另一个重要函数是 `oes()`，它用于在间歇性需求情况下实现模型中的发生部分。Svetunkov（2022）第13章对此进行了详细讨论，其依据是 Svetunkov 和 Boylan（2019年）。

最后但同样重要的是，`smooth` package 包含大量示例文档，其中提供了几乎所有函数的应用示例。例如，可以在 CRAN 上获取：`https://cran.r-project.org/package=smooth`。

8. `smooth`函数的基准测试

最后，为了展示 `smooth` 函数的工作原理，我们使用 M1（Makridakis等人，1982年）、M3（Makridakis和Hibon，2000年）和 Tourism（Athanasopoulos等人，2011年）竞赛数据进行实验，评估以下七种模型：

1. ADAM ETS——通过 `adam()` 函数估计的 ETS 模型；2. ADAM ARIMA——通过 `adam()` 函数估计的 ARIMA 模型。我们将 `model="NNN"` 设为关闭模型中的 ETS 部分，并使用以下命令设置要检查的最大 ARIMA 阶数：`order=list(ar=c(3,2), i=c(2,1), ma=c(3,2), select=TRUE)`；3. ES——通过 `es()` 函数实现的 ETS 模型，该函数只是 `adam()` 的封装；4. SSARIMA——通过 `auto.ssarima()` 估计的状态空间ARIMA模型；5. CES——通过 `smooth` 包中的 `auto.ces()` 函数实现的复指数平滑；6. ETS——通过 `forecast` 包中的 `ets()` 函数实现的 ETS 模型；7. ARIMA——通过 `forecast` 包中的 `auto.arima()` 函数选择的 ARIMA 模型。

由于所考虑的数据集不包含多重季节性时间序列，因此我们没有在实验中纳入 `msarima()`。所有函数的参数均采用默认值。我们针对数据集中每个时间序列，按照原始竞赛所采用的预测时域，对模型不可见的那部分数据生成了预测值。我们为该部分序列生成了点预测和95%预测区间，并使用以下指标评估模型的性能：

- MASE——Hyndman和Koehler（2006年）提出的平均绝对尺度误差；
- RMSSE – Makridakis 等人（2022年）提出的均方根尺度平方误差；

- 覆盖率 – 留出集中落在生成的95%预测区间内的观测值百分比;

- Makridakis 等人（2022）提出的缩放平均区间得分（sMIS）;

- 时间（Time）——估计和为每个序列生成预测值所花费的计算时间（以秒为单位）。

对于平均绝对缩放误差（MASE）、均方根缩放误差（RMSSE）、对称化平均区间分数（sMIS）和时间，数值越低越好。对于覆盖率，数值越接近名义95%，效果越好。

该实验的结果汇总于表1。请注意，由于某些函数的预测值依赖于模拟，结果可能会因运行次数不同而有所变化。

	平均绝对缩放误差	均方根缩放误差	覆盖率	对称化平均区间分数	Time
ADAM ETS	**2.222**	**1.935**	*0.885*	**2.122**	*0.386*
ES	*2.224*	*1.939*	**0.898**	*2.196*	0.477
CES	2.271	1.958	0.812	3.465	**0.236**
ETS	2.263	1.970	0.882	2.258	0.409
自回归积分滑动平均模型	2.300	1.987	0.834	3.007	1.425
ADAM 自回归积分滑动平均模型	2.371	2.048	0.843	3.126	1.376
季节性自回归积分滑动平均模型	2.480	2.133	0.802	3.356	1.811

表1：在 M1、M3 和 Tourism 竞赛数据上评估的各模型的误差指标，采用均值进行汇总。**粗体** 表示表现最佳的模型，而斜体 表示表现次佳的模型。

从表1可以看出，ADAM ETS 在平均绝对缩放误差、均方根缩放误差和对称化平均区间分数方面均优于其他所有模型，不过它与其他 ETS 实现之间的差距似乎并不明显。值得注意的是，它的运行速度略慢于 CES。 `forecast` package 中的 ETS 在这些数据集上的表现略逊于 `smooth` 实现。比较各个自回归积分滑动平均模型实现， `auto.arima()` 的准确度高于 ADAM ARIMA 和季节性自回归积分滑动平均模型，速度也更快，不过它仍不及 ETS 模型。但要注意，自回归积分滑动平均模型的覆盖率低于 ADAM ARIMA，运行速度也更慢。

此示例表明，所开发的函数运行高效，并且可以应用于各种各样的时间序列。表1总结了总体平均表现，但这并不意味着获胜的

模型始终表现最佳。它们的表现会因序列而异，在某些情况下，本实验中表现不佳的模型可能会表现得好得多（例如，正如 Svetunkov 和 Boylan（2020年）所讨论的那样，季节性自回归积分滑动平均模型在历史较短的供应链数据上表现非常出色）。

9. 结论

本文探讨了 R 语言 `smooth` package 中所实现模型背后的理念。函数所使用的状态空间模型不同于传统模型，因此能够引入更多成分，并高效地使用更复杂的模型。我们讨论了在这一框架下如何实现指数平滑状态空间模型和自回归积分滑动平均模型，以及分析人员可以借助它们实现什么。最后，我们通过 M1、M3 和 Tourism 旅游预测竞赛数据的示例，展示了 `smooth` functions 中所实现模型的表现。

本文仅介绍了这一框架、模型和函数。如前所述， `smooth` functions 的主要理念是为研究人员提供灵活性。若读者希望进一步了解这一框架，建议阅读 Svetunkov（2022）的在线专著，并学习 R 语言 `smooth` package 的示例文档中的示例（Svetunkov，2023）。

参考文献

Assimakopoulos, V., Nikolopoulos, K., 2000年。《Theta 模型：一种用于预测的分解方法》。《国际预测学期刊》16, 521–530。
Athanasopoulos, G., Hyndman, R. J., Song, H., Wu, D. C., 2011年。《旅游预测竞赛》。《国际预测学期刊》27 (3), 822–844。Box, G., Jenkins, G., 1976年。《时间序列分析：预测与控制》。Holden-day，加利福尼亚州奥克兰。De Livera, A. M., Hyndman, R. J., Snyder, R. D., 2011年。《利用指数平滑预测具有复杂季节性模式的时间序列》。《美国统计学会杂志》106 (496), 1513–1527。

Engle, R. F., 1982年7月。《英国通货膨胀方差估计的自回归条件异方差》。《计量经济学》50 (4), 987。Gallego, J. L., 2021年。tfarima：传递函数和ARIMA模型。R 程序包版本 0.2.1。URL https://CRAN.R-project.org/package=tfarimaHyndman, R. J., Khandakar, Y., 2008年。《自动时间序列预测：用于 R 的 forecast 包》。《统计软件杂志》26 (3), 1–22。Hyndman, R. J., Koehler, A. B., 2006年。《重新审视预测准确度的度量》。《国际预测学期刊》22 (4), 679–688。Hyndman, R. J., Koehler, A. B., Ord, J. K., Snyder, R. D., 2008年。《指数平滑预测》. 施普林格·柏林·海德堡。Hyndman, R. J., Koehler, A. B., Snyder, R. D., Grose, S., 2002年。《利用指数平滑方法进行自动预测的状态空间框架》。《国际预测学期刊》18 (3), 439–454。Kaluzny, S., TIBCO Software Inc., 2021年。robustarima：稳健 ARIMA 建模。R 程序包版本 0.2.6。URL https://CRAN.R-project.org/package=robustarimaKoehler, A. B., Snyder, R. D., Ord, J. K., Beaumont, A., 2012。《指数平滑预测方法中的离群值研究》。《国际预测学期刊》28 (2), 477–484。Makridakis, S., Andersen, A. P., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E., Winkler, R. L., 1982年。《外推（时间序列）方法的准确度：预测竞赛结果》。《预测杂志》1 (2), 111–153。Makridakis, S., Hibon, M., 2000年。《M3竞赛：结果、结论与启示》。《国际预测学期刊》16, 451–476。Makridakis, S., Spiliotis, E., Assimakopoulos, V., 2020年。《M4竞赛：100,000 个时间序列和 61 种预测方法》。《国际预测学期刊》36 (1), 54–74。

Makridakis, S., Spiliotis, E., Assimakopoulos, V., 2022年10月。《M5准确率竞赛：结果、发现与结论》。《国际预测学期刊》38 (4), 1346–1364。O'Hara-Wild, M., Hyndman, R., Wang, E., 2021。fable：整洁时间序列预测模型。R 程序包版本 0.3.1。URL https://CRAN.R-project.org/package=fableR Core Team, 2022。R：一种用于统计计算的语言与环境。R 统计计算基金会，维也纳，奥地利。URL https://www.R-project.org/Sagaert, Y., Svetunkov, I., 2022。Trace 前向逐步法：瞬间自动选择变量。Snyder, R. D., 1985。动态线性模型的递归估计。《皇家统计学会学报 B 辑（方法论）》47 (2), 272–276。Svetunkov, I., 2022。使用 adam 进行预测与分析。专著。Open-Forecast, （版本：2022-04-18）。URL https://openforecast.org/adam/Svetunkov, I., 2023。smooth：使用状态空间模型进行预测。R 程序包版本 3.2.0。URL https://github.com/config-i1/smoothSvetunkov, I., Boylan, J., 2019。间歇性时间序列的乘法状态空间模型。Svetunkov, I., Boylan, J. E., 2020。用于供应链预测的状态空间 ARIMA。《国际生产研究杂志》58 (3), 818–827。Svetunkov, I., Kourentzes, N., Ord, J. K., 8 2022。复指数平滑。《海军研究后勤》（NRL），31。Svetunkov, I., Petropoulos, F., 2018。老方法，新技巧：简单移动平均的建模视角。《国际生产研究杂志》56 (18), 6034–6047。

Taylor, J. W., 2003。使用双重季节指数平滑进行短期电力需求预测。《运筹学会期刊》54 (8), 799–805。Warren M. Persons, 1919年。一般考虑与假设。《经济学与统计学评论》1 (1), 5–107。Weller, M., Crone, S. F., 2012年11月。供应链预测：最佳实践与基准评估研究。技术报告，兰卡斯特预测中心。Wickham, H., Kuhn, M., Vaughan, D., 2022。generics：R 基础方法未提供的、与模型拟合相关的常用 S3 泛型。R 程序包版本 0.1.2。URL https://CRAN.R-project.org/package=generics