张敬信-《数学建模：算法与编程实现》配套课件

第 11 讲　时间序列分析 I：确定性分解

张敬信

2022 年 5 月 18 日

哈尔滨商业大学

一. 时间序列数据结构

- 为了研究某一事件的规律，依据时间发生的顺序将事件在多个时刻的数值记录下来，就构成了一个**时间序列**，用 $\{y_t\}$ 表示。
- 例如，国家或地区的年度财政收入，股票市场的每日波动，气象变化，工厂按小时观测的产量等。另外，随温度、高度等变化而变化的离散序列，也可以看作时间序列。
- 根据时间序列自身的变化规律，利用外推机制描述和预测时间序列将来的发展趋势，就是**时间序列分析**。
- 时间序列分析的三种经典算法：**确定性分解**、**指数平滑**、ARIMA **模型**。

- `base R` 下的 `ts` 数据类型是专门为时间序列设计的，本质上是一个数值型向量，扩展了时刻属性使得每个数都有一个时刻与之对应。
- 用 `ts(data, start, end, frequency, ...)` 生成时间序列

参数 `frequency` 设置时间频率，默认为 1，表示一年有 1 个数据，`frequency=12`（月度数据），`frequency=52`（周度数则），`frequency=365`（日度数据）。

```r
ts(data = 1:10, start = 2010, end = 2019)     # 年度数据
#> Time Series:
#> Start = 2010
#> End = 2019
#> Frequency = 1
#>  [1]  1  2  3  4  5  6  7  8  9 10
ts(data = 1:10, start = 2010, frequency = 4)  # 季度数据
#>      Qtr1 Qtr2 Qtr3 Qtr4
#> 2010    1    2    3    4
#> 2011    5    6    7    8
#> 2012    9   10
```

tsibble

- fpp3 生态下的 `tsibble` 包提供了整洁的时间序列数据结构 `tsibble`.
- 时间序列数据，无非就是指标数据 + 时间索引（或者再 + 分组索引）[1]
- 对于分组时间序列数据，首先是一个数据框，若有分组变量需采用”长格式”作为一列，只需要指定时间索引、分组索引，就能变成时间序列数据结构。

[1]多元时间序列，就是包含多个指标列.

- 现有 `tibble` 格式的 3 个公司 2017 年的日度股票数据，其中存放 3 只股票的 `Stock` 列为分组索引：

```
library(fpp3)
load("datas/stocks.rda")
stocks
#> # A tibble: 753 x 3
#>   Date       Stock  Close
#>   <date>     <chr>  <dbl>
#> 1 2017-01-03 Google  786.
#> 2 2017-01-03 Amazon  754.
#> 3 2017-01-03 Apple   116.
#> 4 2017-01-04 Google  787.
#> # ... with 749 more rows
```

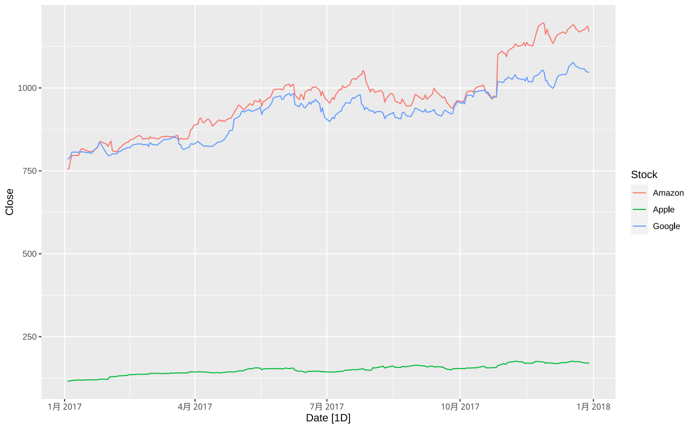
- 用 `as_tsibble()` 将数据框转化为时间序列对象 `tsibble`, 只需要指定时间索引 (index)、分组索引 (key):

```
stocks = as_tsibble(stocks, key = Stock, index = Date)
stocks
#> # A tsibble: 753 x 3 [1D]
#> # Key:       Stock [3]
#>   Date       Stock  Close
#>   <date>     <chr>  <dbl>
#> 1 2017-01-03 Amazon  754.
#> 2 2017-01-04 Amazon  757.
#> 3 2017-01-05 Amazon  780.
#> 4 2017-01-06 Amazon  796.
#> # ... with 749 more rows
```

- tsibble 对象非常便于后续处理和探索:

```
stocks %>%
  group_by_key() %>%
  index_by(weeks = ~ yearweek(.x)) %>%    # 周度汇总
  summarise(avg_week = mean(Close))
#> # A tsibble: 156 x 3 [1W]
#> # Key:       Stock [3]
#>   Stock     weeks avg_week
#>   <chr>    <week>    <dbl>
#> 1 Amazon 2017 W01     772.
#> 2 Amazon 2017 W02     805.
#> 3 Amazon 2017 W03     809.
#> 4 Amazon 2017 W04     830.
#> # ... with 152 more rows
```

```r
autoplot(stocks)        # 可视化
```



1. **差分与延迟**

- **一阶差分**：$\Delta y_t = y_t - y_{t-1}$, 长度为 $T-1$.
- **二阶差分**：$\Delta^2 y_t = \Delta y_t - \Delta y_{t-1}$, 类似的有更高阶差分。
- **季节差分（m 步差分）**：$y_t^{(m)} = y_t - y_{t-m}$.
- **延迟算子**：作用于时间序列，时间刻度减小 1 个单位（序列左移一位）

$$\mathrm{L} y_t = y_{t-1}, \quad \mathrm{L}^2 y_t = y_{t-2}, \cdots$$

2. 平稳性检验

(宽) 平稳性：时间序列的主要性质近似稳定，即统计性质只要保证序列的二阶矩平稳。

平稳性检验：

- **时序图检验**：若无明显的趋势性和周期性，则平稳
- **单位根检验**：通过检验时间序列自回归特征方程的特征根是在单位圆内（平稳）还是在单位圆及单位圆外（非平稳），通常用 ADF 检验或 KPSS 检验

非平稳序列的平稳化处理:

- 若时间序列呈线性趋势，均值不是常数，利用一阶差分将产生一个平稳序列;
- 若时间序列呈二次趋势，均值不是常数，利用二阶差分将产生一个平稳序列;
- 若时间序列的波动呈越来越大趋势，即方差不是常数，通常可利用取对数或开 n 次根号或 Box-Cox 变换转化为平稳序列;
- 若时间序列呈现 “相对环” 趋势，通常将数据除以同时发生的时间序列的相应值转化为平稳序列;
- 先用某函数大致拟合原始数据，再用 ARIMA 模型处理剩余量。

案例：2001 年 10 月—2016 年 8 月出口额数据

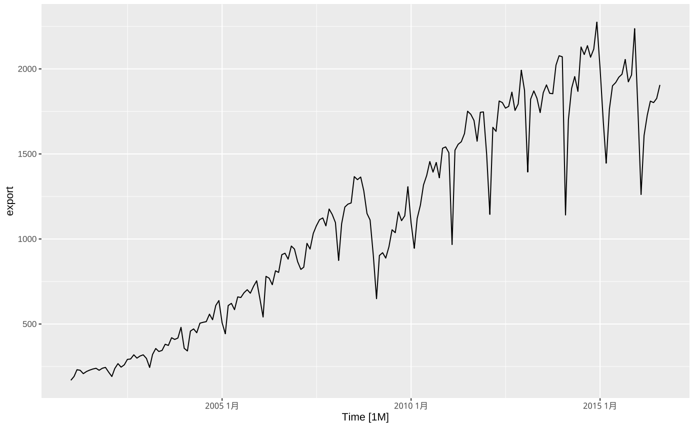
```
library(tidyverse)
library(lubridate)
df = readxl::read_xlsx("datas/export_datas.xlsx")
df
#> # A tibble: 188 x 2
#>   Time        export
#>   <chr>        <dbl>
#> 1 2001 年 10 月   228.
#> 2 2001 年 11 月   240.
#> 3 2001 年 12 月   245.
#> 4 2001 年 1 月    169.
#> # ... with 184 more rows
```

- 转化为 tsibble 对象

```
df = df %>%
  mutate(Time = ymd(str_c(Time, "1 日")) |> yearmonth()) %>%
  as_tsibble(index = Time)
df
#> # A tsibble: 188 x 2 [1M]
#>        Time export
#>       <mth>  <dbl>
#> 1 2001 1 月   169.
#> 2 2001 2 月   192.
#> 3 2001 3 月   231.
#> 4 2001 4 月   228.
#> # ... with 184 more rows
```

- 绘制时序图

```
autoplot(df, export)
```



显然该时间序列非平稳：既有向上的线性趋势，波动（方差）又越来越大。

采用非平稳变平稳的方法：

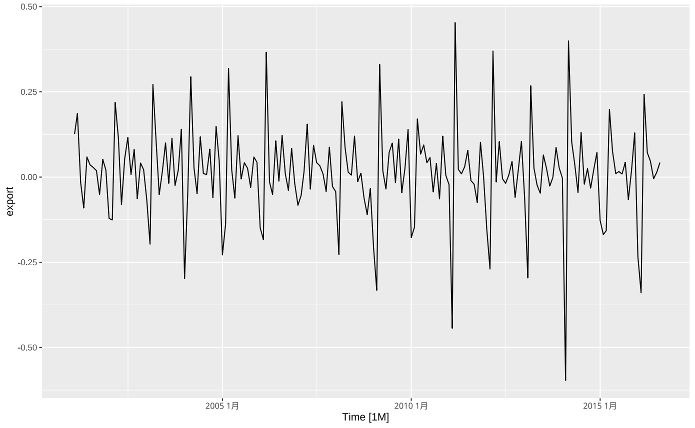
- 线性趋势，做 1 阶差分
- 方差越来越大，取对数[2]

```
df1 = df %>%
  mutate(export = log(export))
df2 = df1 %>%
  mutate(export = difference(export))
```

[2]更好的做法是用 `box-cox` 变换代替取对数.

- 再来看时序图

```
autoplot(df2, export)
```



- 平稳性检验

```
df2 %>%
  features(export, unitroot_kpss)
#> # A tibble: 1 x 2
#>   kpss_stat kpss_pvalue
#>       <dbl>       <dbl>
#> 1     0.191         0.1
```

P 值为 $0.1 > 0.05$, 接受原假设，时间序列平稳。

时间序列分析的一般步骤：

（1）平稳性分析：时序图观察、平稳性检验

（2）时间序列建模

- 若平稳，依据 ACF、PACF 的截尾、拖尾情况进行模型定阶（也可以自动定阶），以确定选择哪种模型：AR(p)、MA(q)、ARMA(p,q)。
- 若非平稳，则
  - **确定性分析**，方法一是确定性分解，分解成长期趋势、季节变动、随机波动；方法二是进行基于移动平均思想的指数平滑法：简单、Holt 双参数、Winter 线性季节；
  - **随机性分析**：做 d 阶差分直到平稳，构建 ARIMA(p,d,q) 模型；对于季节性时间序列，即存在明显的季节性（周、月、季等周期变化），则需要推广到 SARIMA (p,d,q)×(P,D,Q)s 模型，既考虑时间步的差分、自回归、移动平均，又考虑按季节的差分、自回归、移动平均。

(3) 残差白噪声检验（Ljung-Box 检验）:

$$H_0：\text{序列的}k\text{ 阶自相关系数均为}0\text{，即白噪声}$$

- 若通过检验，则说明已经从序列中提取到充分的模型信息，时间序列建模成功。
- 若未通过检验，这有两种可能：
    - 残差序列可能仍存在显著的自相关性，可以对残差序列信息进行二次提取：建立残差自回归模型；
    - 前面建模实际上是假定残差方差相等，但残差序列也可能具有异方差性 (ARCH 检验)，简单的有规律的异方差（比如方差越来越大），对原序列取对数就可以解决，更复杂的异方差就需要单独建模：GARCH 族模型。

三. 确定性分解

时间序列可认为是受不同影响因素共同影响的叠加效果，故非平稳时间序列可按确定性因素进行简单分解：

- **长期趋势**（T_t）：表现出某种倾向，上升或下降或水平；
- **季节变化**（S_t）：周期固定的波动变化；
- **剩余部分**（R_t）：包括随机波动；

按叠加方法的不同分为：

- **加法模型**（$y_t = T_t + S_t + R_t$）：适合趋势、周期的变化幅度不随时间变化。
- **乘法模型**（$y_t = T_t S_t R_t$）：适合趋势、周期的变化幅度随时间变化。

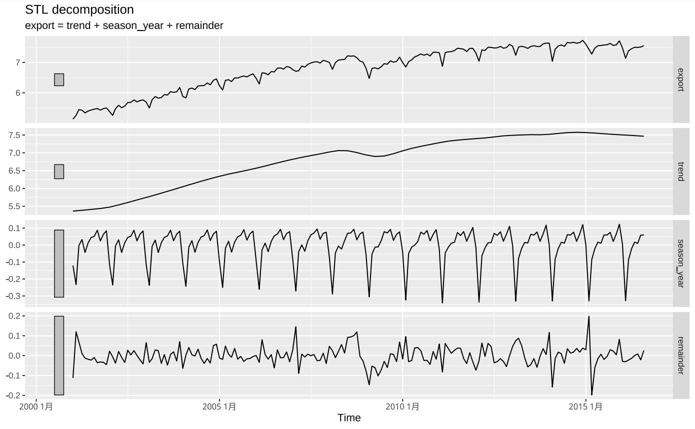
注：适合乘法模型的时间序列，等价于取对数再用加法模型。

案例继续：出口额数据确定性分解建模

fpp3 生态提供了统一的建模框架，STL() 用于确定性分解建模。

```
dcmp = df1 %>%                         # 对平稳方差数据
  model(stl = STL(export))
components(dcmp)
#> # A dable: 188 x 7 [1M]
#> # Key:     .model [1]
#> # :        export = trend + season_year + remainder
#>   .model     Time export trend season_year remainder seaso
#>   <chr>     <mth>  <dbl> <dbl>       <dbl>     <dbl>
#> 1 stl    2001 1 月   5.13  5.36    -0.120     -0.112
#> 2 stl    2001 2 月   5.26  5.37    -0.233      0.120
#> 3 stl    2001 3 月   5.44  5.38    -0.00315    0.0687
#> 4 stl    2001 4 月   5.43  5.39     0.0327     0.0104
#> # ... with 184 more rows
```

```r
components(dcmp) %>%
  autoplot()
```



从上到下，依次是原始值、长期趋势、季节变动、随机噪声。

确定性分解之后如何预测？

分别对各部分进行建模预测：

- 对趋势进行线性回归或曲线拟合，进而往前预测
- 按季节部分的周期性规律，往前预测
- 按剩余部分的随机规律，往前预测

再将三个部分的预测值，加和或乘积（取决于加法模型或乘积模型）得到原序列的预测值。

注： 时间序列的确定性分解，更高级的算法还有 SEATS 和 X11 等。

四. 指数平滑法

指数平滑法进行预测，就是对过去观测值做加权平均，随着观测值的远去，权重呈指数衰减。换句话说，观测越近，相应的权重越大。

另外，在加权时还需要分别考虑序列的水平部分、趋势部分、季节部分，各部分可按加法、乘法形式合成为总预测。

1. 简单指数平滑

适用于没有明确趋势或季节模式（即平稳）的时间序列。

简单指数平滑的加权平均形式表示为：

$$\hat{y}_{T+1|T} = \alpha y_T + (1-\alpha)\hat{y}_T = \cdots = \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots$$

其中，$\alpha \in [0,1]$ 称为水平平滑参数。即 $T+1$ 时刻的预测值是所有观测序列 $y_1, \cdots, y_T$ 的加权平均，权重递减的速率由参数 α 控制。

简单指数平滑也可以写为分量形式（方便推广）：

预测方程：$\hat{y}_{t+h|t} = \ell_t$

水平方程：$\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$

其中，ℓ_t 表示 t 时刻的水平估计值。

2. Holt 线性指数平滑

Holt 线性指数平滑是简单指数平滑法的推广，适合带趋势的时间序列。其分量形式表示为：

预测方程：$\hat{y}_{t+h|t} = \ell_t + hb_t$

水平方程：$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$

趋势方程：$b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1}$

其中，$\alpha \in [0,1]$ 为水平平滑系数，$\beta \in [0,1]$ 为趋势平滑系数；ℓ_t 表示 t 时刻的水平估计值，b_t 表示 t 时刻的趋势估计值。

3. Holt-Winters 季节指数平滑

Holt-Winters 季节指数平滑是 Holt 线性趋势法的推广，适合带趋势、季节 (周期）性的时间序列。

又增加一个季节方程，用 m 表示季节频率，即一年中包含的季节数，比如季度数据 $m = 4$, 月度数据 $m = 12$.

季节性加入模型的方式有两种：

- 当季节变化在该时间序列中大致保持不变时，通常选择加法模型
- 当季节变化与时间序列的水平成比例变化时，通常选择乘法模型

Holt-Winters **加法模型**：

$$
\begin{aligned}
\text{预测方程：}\ & \hat{y}_{t+h|t} = \ell_t + hb_t + s_{t+h-m(k+1)} \\
\text{水平方程：}\ & \ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
\text{趋势方程：}\ & b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \\
\text{季节方程：}\ & s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}
\end{aligned}
$$

其中，$k = [(h-1)/m]$, $\gamma \in [0,1]$ 为季节平滑参数。

- **水平方程**表示 t 时刻的水平预测值 ℓ_t，为季节调整的观测值 $y_t - s_{t-m}$ 与非季节性预测值 $(\ell_{t-1} + b_{t-1})$ 的加权平均；
- **趋势方程**表示 t 时刻的趋势预测值 b_t，为当前趋势值 $\ell_t - \ell_{t-1}$ 与上一期的趋势估计值 b_{t-1} 的加权平均；
- **季节方程**表示 t 时刻的季节预测值 s_t，为当前季节指数 $(\ell_{t-1} + b_{t-1})$ 与去年同一季节（即 m 个时刻前）季节指数 s_{t-m} 的加权平均。

Holt-Winters **乘法模型**:

$$
\begin{aligned}
\text{预测方程：} & \hat{y}_{t+h|t} = (\ell_t + hb_t)s_{t+h-m(k+1)} \\
\text{水平方程：} & \ell_t = \alpha \frac{y_t}{s_{t-m}} + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
\text{趋势方程：} & b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \\
\text{季节方程：} & s_t = \gamma \frac{y_t}{\ell_{t-1} + b_{t-1}} + (1-\gamma)s_{t-m}
\end{aligned}
$$

fpp3 生态提供了统一的建模框架，ETS() 用于指数平滑建模，模型公式右端通过 error()，trend()，season() 设置随机部分、趋势部分、季节部分以何种方式加入模型，“N” 表示不加入模型，“A” 表示加法形式，“M” 表示乘法形式。

- 拟合 Holt-Winters 季节指数平滑模型

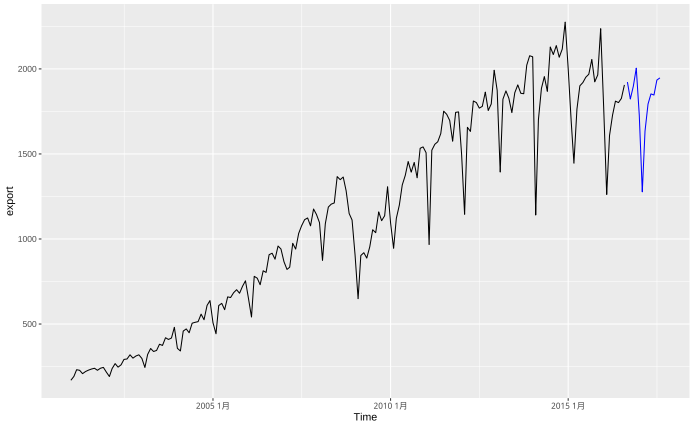
```
fit = df %>%
  model(add = ETS(export ~ error("M") + trend("A")
                  + season("M")))
glance(fit)
#> # A tibble: 1 x 9
#>   .model  sigma2 log_lik   AIC  AICc   BIC   MSE   AMSE  <
#>   <chr>    <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>  <
#> 1 add    0.00535  -1271. 2577. 2580. 2632. 9035. 10793. 0.
```

- 模型预测，预测未来 12 期：

```
pred = forecast(fit, h = 12)
pred
#> # A fable: 12 x 4 [1M]
#> # Key:     .model [1]
#>   .model      Time          export .mean
#>   <chr>      <mth>          <dist> <dbl>
#> 1 add     2016 9 月  N(1923, 19768) 1923.
#> 2 add    2016 10 月  N(1823, 21881) 1823.
#> 3 add    2016 11 月  N(1899, 28329) 1899.
#> 4 add    2016 12 月  N(2004, 36847) 2004.
#> # ... with 8 more rows
```

- 可视化原时间序列及预测结果：

```r
autoplot(pred, df, level = NULL)
```



本篇主要参阅 (张敬信, 2022), (Hyndman and Athanasopoulos, 2021), 以及包文档，模板感谢 (黄湘云, 2021), (谢益辉, 2021).

参考文献

Hyndman, R. J. and Athanasopoulos, G. (2021). *Forecasting: Principles and Practice*. O Texts, 3 edition.

张敬信 (2022). *R 语言编程：基于 tidyverse*. 人民邮电出版社, 北京.

谢益辉 (2021). *rmarkdown: Dynamic Documents for R*.

黄湘云 (2021). *Github: R-Markdown-Template*.