

## 数据

- 1. 季度数据采用: (季度财报) 基本每股收益, 并收集“是否发生差错更正” [IfCorrect] 与“差错更正披露日期”[DeclareDat]数据
- 2. 日度数据采用: (收盘价的报酬率)
- 3. 应计算得所数据:
  - 季度财报估计值 - 季度财报公布值 => 季度财报预估值与真实值之间差
  - 季度财报公布时间 - 季度末时间 => 季度财报滞后公布时间
    - 股票从季度末至公布时间 (滞后时间区间) 中的股票变化 ( $\frac{\text{初}-\text{末}}{\text{末}}$ ?)

论文阅读[季度财报公告时间对股票收益影响的实证研究](#)所得信息:

基于实际季度收益与预测季度收益的比较, 计算了**每个公司每个季度的收益预测误差**

- 数据的获取:  
数据应满足以下条件:
  - 1. 公司的每日股价数据应是可获得的
  - 2. 公司的季度财报公布日期需要是已知的
  - 3. 公司的季度收益数据应是可获得的
- 另外, 文中用到的是公告滞后预测模型, 用于预测从季度末到公司*i*在季度*q*的财报公布的天数 (即财报滞后公布天数)

## 方法

- 总链接: [CRAN - Package midasr \(r-project.org\)](https://cran.r-project.org/web/packages/midasr/index.html)
- github-MIDASr包: [mpiktas/midasr](https://github.com/mpiktas/midasr): [用于混合频率时间序列数据分析的R包](#)  
midasr包的目标是估计具有参数函数约束的混合频率模型。

论文(R包) [Mixed Frequency Data Sampling Regression Models-The R Package midasr.pdf](#):

安装:

```
install.packages("devtools", type = "win.binary")
install_github("midasr","mpiktas")
```

## 常用包

- midas\_r ——使用NLS进行MIDAS回归估计。
- midas\_nlpr ——用非线性参数MIDAS回归估计。
- midas\_sp ——半参数和部分线性MIDAS回归。
- midas\_gr ——分位数MIDAS回归。
- mls ——时间序列嵌入到较低的频率, 用于指定MIDAS模型的灵活函数。
- MLSD ——利用可用的日期信息嵌入到较低频率的时间序列。
- hAh.test 和 hAhr.test ——MIDAS回归的充分性测试。
- forecast ——预测MIDAS回归。
- midasr\_ic\_table ——使用信息标准进行滞后项的选择。

average\_forecast — 计算加权预测组合。  
Select\_and\_forecast —— 执行模型选择， 然后使用所选的模型进行预测。

该包提供了通用函数的常用方法， 可用于拟合的MIDAS回归对象: summary、 coef、 residuals、 deviance、 fitted、 predict、 loglik。  
它还有其他的方法来估计稳健的标准误差: estfun 和 bread  
该包还提供了所有流行的MIDAS回归限制， 如归一化Almon指数、 归一化beta等。

数据处理

MIDAS回归模型的关键特殊性在于， 在不同频率下观察到的变量的观察长度不同， 需要如第2节所述进行对齐。 目前没有执行这种转换的R函数， 包midasr为这些挑战提供了解决方案。 函数 fmls(x, k, m 精确地执行了在公式4中定义的转换， 将给定(潜在的)高频序列的观测向量x转换为(k + 1)低频序列(同时有k个滞后向量:k是高频特征的个数)的相应叠加矩阵， 该矩阵由最大滞后阶数k和频率倍差m定义。

公式4:

$$\begin{bmatrix} y_l \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} y_{l-1} & \cdots & y_{l-p} \\ \vdots & \vdots & \vdots \\ y_{n-1} & \cdots & y_{n-p} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_p \end{bmatrix} + \sum_{i=0}^k X^{(i)} \begin{bmatrix} \beta_0^{(i)} \\ \vdots \\ \beta_l^{(i)} \end{bmatrix} + \begin{bmatrix} \varepsilon_l \\ \vdots \\ \varepsilon_n \end{bmatrix},$$

where

$$X^{(i)} := \begin{bmatrix} x_{um_i}^{(i)} & x_{um_i-1}^{(i)} & \cdots & x_{um_i-l}^{(i)} \\ x_{(u+1)m_i}^{(i)} & x_{(u+1)m_i-1}^{(i)} & \cdots & x_{(u+1)m_i-l}^{(i)} \\ \vdots & \vdots & \cdots & \vdots \\ x_{tm_i}^{(i)} & x_{tm_i-1}^{(i)} & \cdots & x_{tm_i-l}^{(i)} \\ \vdots & \vdots & \cdots & \vdots \\ x_{(n-1)m_i}^{(i)} & x_{(n-1)m_i-1}^{(i)} & \cdots & x_{(n-1)m_i-l}^{(i)} \\ x_{nm_i}^{(i)} & x_{nm_i-1}^{(i)} & \cdots & x_{nm_i-l}^{(i)} \end{bmatrix}, \tag{4}$$

and  $u$  is the smallest integer such that  $um_i - l > 0$  and  $u > p$ .

数据处理的基本函数

Function	Description	Example	Notes
m1s(x, k, m)	Stacks a HF data vector $x$ into a corresponding matrix of observations at LF of size $\frac{\dim x}{m} \times \dim k$ : from the first to the last HF lag defined by vector $k$ .	m1s(x, 2:3, 3)	$\frac{\dim x}{m}$ must be an integer (NA are allowed). For $m = 1$ , the function produces lags of $x$ that are defined by vector argument $k$ , e.g., m1s(x, 2:3, 1) yields a data set containing the lags $x_{t-2}$ and $x_{t-3}$ of $x_t$ .
fmls(x, k, m)	Same as m1s, except that $k$ is a scalar and the $k + 1$ lags are produced starting from 0 up to $k$ .	fmls(x, 2, 3)	fmls(x, 2, 3) is equivalent to m1s(x, 0:2, 3).
dmls(x, k, m)	Same as fmls, only the resulting matrix contains $k + 1$ first-order HF differences of $x$ .	dmls(x, 2, 3)	m1s(x, 1, 1) can be used in dmls to get stacking of lagged differences, e.g., dmls(m1s(x, 1, 1), 2, 3).

Table 2: A summary of a basic data handling functionality in the package midasr.

函数	描述	例子	注释
m1s(x,k,m)	x为向量 / k为滞后阶数组成的向量， 零代表没有滞后， 是同期向量 / m表示倍差	m1s(x,2:3,3)	$\frac{\dim x}{m}$ 必须为整数(允许为NA空值)。对于m = 1， 函数产生的滞后项x， 由向量参数k定义。例如： m1s(x, 2:3, 1) 产生一个包含滞后项 $x_t$ 的 $x_{t-2}$ 和 $x_{t-3}$ 的数据集 合:m1s(x,2:3,1) → { $x_{t-2}, x_{t-3}$ }。

函数	描述	例子	注释
<code>fmls(x,k,m)</code>	x为向量 / k为最大滞后阶数 / m频率倍差 【与mls相同，只是k是一个标量，从0到k产生k +1个滞后】	<code>fmls(x,2,3)</code>	<code>fmls(x, 2,3)</code> 与 <code>mls(x,0:2,3)</code> 等价
<code>dmls(x,k,m)</code>	x为向量 / k为最大滞后阶数 / m频率倍差 【与fmls相同，只是得到的矩阵包含x的k+ 1个一阶HF差分】	<code>dmls(x,2,3)</code>	<code>mls (x, 1,1)</code> 可以使用 <code>dmls</code> , 得到滞后差异的叠加，例如， <code>dmls(mls(x,1,1),2,3)</code>

直接看函数解释有点困难，还是得结合实际实验

2. 安装R环境与Rstudio，并对midasr中数据处理常用函数进行测试

