

统计软件与金融数据

第五章日期与时间

罗智超 (*ROKIA.ORG*)

Contents

通过本章你将学会	2
日期处理	2
日期时间类	2
日期时间包	3
时间序列作图包的简单历史	4
日期时间处理包	4
日期时间其他资源链接	6
日期时间小结	6

通过本章你将学会

- 日期时间函数
- 日期时间操作

日期处理

日期时间类

- 系统内嵌三种类：Date, POSIXct, POSIXlt.
- Date

This is the class to use if you have only dates, but no times, in your data.

```
## 创建日期

date<-"2016-01-01"

dt1 <- as.Date("2015-01-01")
dt2<-as.Date("2015-12-31")

dt2 <- as.Date("08/20/2012", format = "%m/%d/%Y")
dt3 <- as.Date("October 6, 2010", format = "%B %d, %Y")

as.Date("2015/35/1",format="%Y/%U/%u")

## 查看所有格式:
?strptime

## 日期计算

## 计算间隔
dt1 - dt2
## 加减天数
dt2 + 10
dt2 - 10

## 创建日期向量
dt1 <- as.Date("2015-01-01")
dt2<-as.Date("2015-12-31")

y2015<-seq(from=dt1,to=dt2,by="2 weeks")

six.weeks <- seq(dt1, length = 6, by = "week")
six.weeks <- seq(dt1, length = 6, by = "2 weeks")
six.weeks <- seq(dt1, length = 6, by = 14)

##diff
three.dates <- as.Date(c("2010-07-22", "2011-04-20", "2012-10-06"))
diff(three.dates)
```

```
difftime(dt1, dt2, units = "weeks")

## 日期-数值转换
a<-as.Date("2014-01-03")
b<-as.integer(a)
c<-as.Date(b,origin="1960-01-01")#SAS date
```

- POSIXct

如果要包含时间信息，就要使用 POSIXct 类

```
## 创建 POSIXct 类
tm1 <- as.POSIXct("2013-07-24 23:55:26")
tm2 <- as.POSIXct("25072013 08:32:07", format = "%d%m%Y %H:%M:%S")
## 设定时区
tm3 <- as.POSIXct("2010-12-01 11:42:03", tz = "GMT")
## 时间计算
## 加减 30 秒
tm1 + 30
tm1 - 30
## 时间比较
tm2 - tm1

## 获取当前时间 (默认是 POSIXct 格式)
```

- POSIXlt

该类以 List 类型存储日期时间，更方便提取元素，“ct”代表 calendar time 日历时间，lt 代表 local time.

```
## 创建时间
tm1.lt <- as.POSIXlt("2013-07-24 23:55:26")
unclass(tm1.lt)
unlist(tm1.lt)
## 提取元素

tm1.lt$sec
tm1.lt$yday

## 截取长度 (日期、分钟)
trunc(tm1.lt, "days")
trunc(tm1.lt, "mins")
```

日期时间包

- chron

如果不需要考虑时区和 daylight savings time，可以考虑使用 chron 类，需要调用 library(chron)

```
library(chron)
tm1.c <- as.chron("2013-07-24 23:55:26")
tm2.c <- as.chron("07/25/13 08:32:07", "%m/%d/%y %H:%M:%S")
dates(tm1.c)
tm2.c > tm1.c
tm1.c + 10
tm2.c - tm1.c
difftime(tm2.c, tm1.c, units = "hours")
```

- ts
- zoo

zoo 是一个 R 语言类库，zoo 类库中定义了一个名为 zoo 的 S3 类型对象，用于描述规则的和不规则的有序的时间序列数据。zoo 对象是一个独立的对象，包括索引、日期、时间，只依赖于基础的 R 环境，zoo 是时间序列的基础，也是股票分析的基础。

```
## 构建一个 zoo 对象，以时间为索引
library(zoo)
x.Date <- as.Date("2003-02-01") + c(1, 3, 7, 9, 14) - 1
x.Date
class(x.Date)
x1 <- zoo(rnorm(5), x.Date)
class(x1)
x.Time<-as.POSIXct("2013-07-24 23:55:26")+c(1, 3, 7, 9, 14) - 1
x2<-zoo(rnorm(5),x.Time)
class(x2)
plot(x2)
## 修改 zoo 的数据部分
coredata(x2)
## 修改 zoo 的索引部分
index(x2)
## 按时间过滤 window.zoo
window(x, start = as.Date(""), end = as.Date(""))
## 合并多个 zoo
merge(x1, x2, all = FALSE)

## 数据滚动处理 rollapply
z <- zoo(11:15, as.Date(31:35))
rollapply(z, 2, mean)
```

- xts

*zoo 作为时间序列的基础库，是面向通用的设计，可以用来定义股票数据，也可以分析天气数据。但由于业务行为的不同，我们需要更多的辅助函数，来帮助我们更高效的完成任务。xts 扩展了 zoo，提供更多的数据处理和数据变换的函数。

*xts 对象是金融时间序列的标准。quantmod getSymbols() 的提取结果就是一个 xts 对象。wind 的数据提取结果是一个 list，其中的 data 部分包括了日期时间信息和数据，需要转换成 xts 对象。

时间序列作图包的简单历史

- | | Package | Date |
|-----|----------------------|------------|
| 1 | ts | 1999-08-27 |
| 2 | lattice/grid | 2002-04-29 |
| 3 | zoo | 2004-10-08 |
| 4 | zoo/lattice | 2006-07-06 |
| 5 | PerformanceAnalytics | 2007-02-02 |
| 6 | ggplot2 | 2007-06-10 |
| 7 | quantmod/TTR | 2007-10-07 |
| 8 | xts | 2008-02-17 |
| 9 | timeSeries | 2009-05-17 |
| 10 | xtsExtra | 2012-05-30 |
| *11 | rCharts | 2013-04-10 |

日期时间处理包

- lubridate

* 这个包在 POSIXct 的基础上增加了很多日期时间操作函数

```
## 创建日期时间
## 默认使用 UTC 时区
tm1.lub <- ymd_hms("2013-07-24 23:55:26")
```

```

tm2.lub <- mdy_hm("07/25/13 08:32")
tm3.lub <- ydm_hm("2013-25-07 4:00am")
tm4.lub <- dmy("26072013")

```

提取日期时间元素

```

year(tm1.lub)
week(tm1.lub)
wday(tm1.lub, label = TRUE)
hour(tm1.lub)
tz(tm1.lub)
second(tm2.lub) <- 7
tm2.lub

```

*Lubridate 区分四种对象: instants, intervals, durations, and periods.

*instants 是一个时点

*Intervals, durations, and periods 是三种区间表示方法

```

## 日期时间都是 instants 对象
is.instant(tm1.lub)
## 日期时间取整
round_date(tm1.lub, "minute")
round_date(tm1.lub, "day")
## 查看在不同时区的时间
with_tz(tm1.lub, "America/Los_Angeles")
## 改变时区
force_tz(tm1.lub, "America/Los_Angeles")

## 日期时间区间
in.bed <- as.interval(tm1.lub, tm2.lub)

## 判断某时间是否在某区间内
tm3.lub %within% in.bed

## 判断时间范围是否重叠
daylight <- as.interval(ymd_hm("2013-07-25 06:03"), ymd_hm("2013-07-25 20:23"))
int_overlaps(in.bed, daylight)

##duration 没有设定开始结尾时间, 只是设定了一个持续时间范围, 以秒为最小单位
## 创建 duration
ten.minutes <- dminutes(10)
five.days <- ddays(5)
one.year <- dyears(1)
as.duration(in.bed)

##period 没有设定开始结尾时间, 只是设定了一个持续时间范围, 与 duration 表现形式不一样
three.weeks <- weeks(3)
four.hours <- hours(4)
sabbatical <- months(6) + days(12)

```

日期时间其他资源链接

- date and time tutorials for R:
- <http://science.nature.nps.gov/im/datamgmt/statistics/r/fundamentals/dates.cfm>
- http://en.wikibooks.org/wiki/R_Programming/Times_and_Dates
 - lubridate:
- <http://www.jstatsoft.org/v40/i03/paper>
 - time zone and daylight saving time info:
- <http://www.timeanddate.com/>
- http://en.wikipedia.org/wiki/List_of_tz_database_time_zones
- <http://www.twinsun.com/tz/tz-link.htm>

日期时间小结

- 如果只有日期，就用 `as.Date`
- 如果有时间，通常 `POSIXct` 是最好选择
- `POSIXlt` 提取日期时间元素最方便
- 如果不需要处理时区和 daylight savings time(类似夏令时),`chron` 也很方便.
- 最好用、使用频率最高的是 `as.Date` 和 `as.xts`
- `strptime` 是个好东西