金融危機預測- HW6

魏上傑

2023-04-03

目錄

1 Currency Crisis and Banking Crisis

1.1 建立日期資料

```
start_date <- as.Date("1970-01-01")
end_date <- as.Date("1998-12-01")
all_dates <- seq(start_date, end_date, by = "month")
data <- data.frame(Date = all_dates)
```

1.2 讀入資料

```
library(readxl)
Currency <- read_excel("Crises.xlsx", sheet="Currency_Crises")
Currency$Date <- as.Date(paste(Currency$Year, "-", Currency$Month, "-01", sep = ""))

Banking <- read_excel("Crises.xlsx", sheet="Banking_Crises")

# 有些國家的銀行危機沒有提供月資料, default 月份為一月
Banking$Month <- ifelse(is.na(Banking$Month), 1, Banking$Month)
Banking$Date <- as.Date(paste(Banking$Year, "-", Banking$Month, "-01", sep = ""))



Currency <- Currency[, c("Date", "Country")]
Currency$currency_crisis <- 1

Banking <- Banking[, c("Date", "Country")]
Banking$banking_crisis <- 1
```

1.3 長表轉寬表

```
library(tidyr)

Currency_wide <- pivot_wider(Currency, names_from = Country,
                             values_from = currency_crisis)
Banking_wide <- pivot_wider(Banking, names_from = Country,
                            values_from = banking_crisis)

Currency_wide <- merge(data, Currency_wide, by = c("Date"), all = TRUE)
Banking_wide <- merge(data, Banking_wide, by=c("Date"), all = TRUE)



Currency_wide[is.na(Currency_wide)] <- 0
Banking_wide[is.na(Banking_wide)] <- 0
```

1.4 寬表轉長表

```
Currency_long <- pivot_longer(Currency_wide, -1,
                              names_to = "Country",
                              values_to = "crisis")
Banking_long <- pivot_longer(Banking_wide, -1,
                             names_to = "Country",
                             values_to = "crisis")

Year <- rep(1970:1998, each=12)
Month <- rep(1:12, 29)



Currency_long <- data.frame(Currency_long, Year, Month)
Banking_long <- data.frame(Banking_long, Year, Month)
```

1.5 計算條件機率函數

給定在時點 t 發生貨幣危機，在時點 t 至時點 t+12 發生銀行危機的機率。(隱含貨幣危機引發銀行危機)

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# 給定時點 t 發生貨幣危機
cond_prob_on_currency <- function(nation, t_year, t_month){

  # 找到指定國家的貨幣危機資料
  currency_nation <- filter(Currency_long, Country==nation,
```

```
                            Year==t_year, Month==t_month)

  if (currency_nation$crisis==0){
    return("There's no currency crisis at given time")
  }

  # 找到指定國家在時間點 t 到 t+12 的銀行危機資料
  banking_nation <- filter(Banking_long, Country == nation,
                           Year == t_year, Month >= t_month)

  if (t_month + 12 > 12) {
    banking_nation <- rbind(banking_nation,
                            filter(Banking_long,
                                   Country == nation,
                                   Year >= t_year + 1,
                                   Month <= t_month + 12 - 12))
  }




  crisis_bank <- filter(banking_nation, crisis == 1)

  # 計算給定國家在時間點 t 發生貨幣危機,
  # 從時間點 t 到時間點 t+12 這段時間發生銀行危機的條件機率
  if (nrow(crisis_bank) == 0) {
    return("There's no banking crisis during the period of time")
  } else {
    prob_bank <- nrow(crisis_bank) / nrow(banking_nation)
    return(prob_bank)
  }


}
```

```
cond_prob_on_currency("Argentina", 1985, 1)
```

```
## [1] "There's no currency crisis at given time"
```

```
cond_prob_on_currency("Argentina", 1986, 2)
```

```
## [1] "There's no banking crisis during the period of time"
```

```
cond_prob_on_currency("Argentina", 1987, 4)
```

```
## [1] 0.01886792
```

給定在時點 t 發生銀行危機，在時點 t 至時點 t+12 發生貨幣危機的機率。(隱含銀行危機引發貨幣危機)

```
library(dplyr)
# 給定時點 t 發生銀行危機
cond_prob_on_banking <- function(nation, t_year, t_month){

  # 找到指定國家銀行危機資料
  banking_nation <- filter(Banking_long, Country==nation,
                           Year==t_year, Month==t_month)

  if (banking_nation$crisis==0){
    return("There's no banking crisis at given time")
  }

  # 找到指定國家在時間點 t 到 t+12 的貨幣危機資料
  currency_nation <- filter(Currency_long, Country == nation,
                            Year == t_year, Month >= t_month)

  if (t_month + 12 > 12) {
    currency_nation <- rbind(currency_nation,
                             filter(Currency_long,
                                    Country == nation,
                                    Year >= t_year + 1,
                                    Month <= t_month + 12 - 12))
  }




  crisis_currency <- filter(currency_nation, crisis == 1)
```

```
  # 計算給定國家在時間點 t 發生銀行危機, 從時間點 t 到時間點 t+12 這段時間發生貨幣危機的條件
  if (nrow(crisis_currency) == 0) {
    return("There's no currency crisis during the period of time")
  } else {
    prob_currency <- nrow(crisis_currency) / nrow(currency_nation)
    return(prob_currency)
  }


}
```

```
cond_prob_on_banking("Argentina", 1976,5)
```

```
## [1] 0.02542373
```

```
cond_prob_on_banking("Argentina", 1983,12)
```

```
## [1] 0.02209945
```

```
cond_prob_on_banking("Argentina", 1992,3)
```

```
## [1] 0.03571429
```

```
cond_prob_on_banking("Brazil", 1989, 2)
```

```
## [1] "There's no banking crisis at given time"
```

2 Housing Price Bubble

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.4
## v ggplot2   3.4.1     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(lubridate)
library(mFilter)
library(readxl)
library(xts)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
##
## ##################################### WARNING #####################################
## # We noticed you have dplyr installed. The dplyr lag() function breaks how    #
## # base R's lag() function is supposed to work, which breaks lag(my_xts).      #
## #                                                                             #
## # Calls to lag(my_xts) that you enter or source() into this session won't     #
## # work correctly.                                                             #
## #                                                                             #
## # All package code is unaffected because it is protected by the R namespace   #
## # mechanism.                                                                  #
## #                                                                             #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                             #
## # You can use stats::lag() to make sure you're not using dplyr::lag(), or you #
## # can add conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop   #
## # dplyr from breaking base R's lag() function.                                #
## ##################################### WARNING #####################################
##
```

```
## Attaching package: 'xts'
##
## The following objects are masked from 'package:dplyr':
##
##     first, last
```

2.1 Including the datas

Data of housing price is the JST database, which can be downloaded Here.

```
JST = read_excel("JSTdatasetR6.xlsx")
head(JST)
```

```
## # A tibble: 6 x 59
##    year country iso     ifs   pop rgdpmad rgdpbarro rconsbarro   gdp    iy   cpi
##   <dbl> <chr>   <chr> <dbl> <dbl>   <dbl>     <dbl>      <dbl> <dbl> <dbl> <dbl>
## 1  1870 Austra~ AUS     193  1775   3273.      13.8       21.4  209. 0.109  2.71
## 2  1871 Austra~ AUS     193  1675   3299.      13.9       19.9  212. 0.105  2.67
## 3  1872 Austra~ AUS     193  1722   3553.      15.0       21.1  227. 0.130  2.54
## 4  1873 Austra~ AUS     193  1769   3824.      16.2       23.3  267. 0.125  2.54
## 5  1874 Austra~ AUS     193  1822   3835.      16.3       23.5  288. 0.142  2.67
## 6  1875 Austra~ AUS     193  1874   4138.      17.6       25.7  301. 0.161  2.75
## # i 48 more variables: ca <dbl>, imports <dbl>, exports <dbl>, narrowm <dbl>,
## #   money <dbl>, stir <dbl>, ltrate <dbl>, hpnom <dbl>, unemp <dbl>,
## #   wage <dbl>, debtgdp <dbl>, revenue <dbl>, expenditure <dbl>, xrusd <dbl>,
## #   tloans <dbl>, tmort <dbl>, thh <dbl>, tbus <dbl>, bdebt <dbl>, lev <dbl>,
## #   ltd <dbl>, noncore <dbl>, crisisJST <dbl>, crisisJST_old <dbl>, peg <dbl>,
## #   peg_strict <dbl>, peg_type <chr>, peg_base <chr>, JSTtrilemmaIV <dbl>,
## #   eq_tr <dbl>, housing_tr <dbl>, bond_tr <dbl>, bill_rate <dbl>, ...
```

2.2 Functions to simplify the process

`interval()` 函數用於創建時間區間物件。時間區間表示某個時間段，例如從 2023 年 1 月 1 日到 2023 年 12 月 31 日這一年的時間區間。該函數可以接受兩個參數：`start` 和 `end`，它們分別表示時間區間的開始日期和結束日期。這些日期必須是 R 中支持的日期格式，例如 `Date`、`POSIXct` 或 `POSIXlt`。

```
# 創建一個時間區間物件
my_interval <- interval(start = as.POSIXct("2023-01-01"), end = as.POSIXct("2023-12-31"))
```

```
# 顯示時間區間的開始和結束日期
my_interval
```

```
## [1] 2023-01-01 CST--2023-12-31 CST
```

ISOdate() 函數用於創建一個日期時間對象。該函數接受多個參數，包括年、月、日、時、分和秒。這些參數必須是數值型別。ISOdate() 函數將返回一個 POSIXct（POSIXct 表示自 “1970-01-01 00:00:00 UTC” 以來的秒數）日期時間對象。使用 ISOdate() 函數時，建議在年份前面加上 “-1” 或 “+1”，以便涵蓋整個時間區間。

```
# 創建一個日期時間對象
my_datetime <- ISOdate(year = 2023, month = 4, day = 1, hour = 14, min = 30, sec = 0)

# 顯示日期時間對象
my_datetime
```

```
## [1] "2023-04-01 14:30:00 GMT"
```

```
year.interval = function(a,b){
  return(
    interval(start=ISOdate(a-1,1,1), end = ISOdate(b+1,1,1))
  )
}
```

2.3 The main code for housing

```
housing.price = JST %>%
  mutate(year = as.Date(ISOdate(year,1,1)) ,
         real_hp = log(hpnom/cpi)) %>%
  select(year, real_hp, iso)
head(housing.price)
```

```
## # A tibble: 6 x 3
##   year       real_hp iso
##   <date>       <dbl> <chr>
## 1 1870-01-01   -1.71 AUS
## 2 1871-01-01   -1.74 AUS
## 3 1872-01-01   -1.66 AUS
```

```
## 4 1873-01-01   -1.69 AUS
## 5 1874-01-01   -1.55 AUS
## 6 1875-01-01   -1.58 AUS
```

```
get.housing.bubble.plot = function(d, title.name, country.to.trend, start, end){

  time.interval = year.interval(start, end)

  # 選取國家，去除 NA 資料
  housing.price.country = d %>%
    filter(iso == country.to.trend, !is.na(real_hp))

  # 選取第一年的數據作為分析的開始年份
  start.year = housing.price.country$year[1]

  # hpfilter() 函數進行 Hodrick-Prescott 濾波
  # ，將房價數據分解為趨勢和週期兩部分。
  hp.trended = housing.price.country$real_hp %>%
    hpfilter(freq = 200, type = "lambda")

  housing.price.country$trend = hp.trended$trend
  housing.price.country$cycle = hp.trended$cycle

  # 選取 housing.price.country 數據框中位於指定時間區間內的數據。
  housing.price.country.interval = housing.price.country %>%
    filter(year %within% time.interval)

  # 計算 housing.price.country 數據框中週期項的標準差。
  std.cycle = housing.price.country %>%
    summarise(sd = sd(cycle)) %>%
    as.numeric()

  housing.price.country.interval = housing.price.country.interval %>%
    mutate(upper = trend + std.cycle,
           lower = trend - std.cycle,
           is.bubble = real_hp > upper)


  p = housing.price.country.interval %>% ggplot(aes(x = year)) +
    geom_ribbon(aes(ymax = upper, ymin = lower), fill='grey70') +
```

```
    geom_line(aes(y=real_hp), col="navyblue")+
    geom_point(aes(x=year, y=real_hp), pch=19 ,
               data=housing.price.country.interval %>%
                 filter(is.bubble))+
    geom_text(aes(x=year, y=real_hp,label=year(year)),
               data=housing.price.country.interval %>%
                filter(is.bubble),nudge_y = std.cycle/3)+
    ylab("")+
    ggtitle(paste(title.name, country.to.trend, paste(start, end, sep = "-"))
            )+
    theme_classic()

  show(p)

  ggsave(filename =  paste(country.to.trend, "_HP.png"),
         p,
         width = 1500, height = 900, units ="px",
         )
}
```

- `d`: 函數的第一個參數，表示數據框（data frame），即存儲數據的表格。
- `title.name`: 函數的第二個參數，表示 Bubble 圖的標題。
- `country.to.trend`: 函數的第三個參數，表示需要進行趨勢分析的國家。
- `start` 和 `end`: 函數的第四和第五個參數，表示需要進行分析的時間區間的開始和結束年份。
- `hpfilter()` 是一個時間序列分析中常用的平滑方法，用於將時間序列分解成趨勢和週期兩個部分。

`hpfilter()` 函數通常接收三个参数:

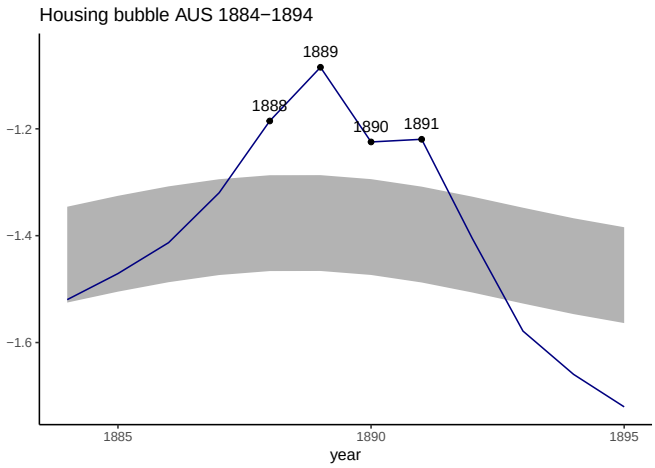
- `x`: 要進行平滑的時間序列數據，可以是一個向量、矩陣或數據框。
- `freq`: 用於計算週期部分的頻率，通常設置為數據中的規律變化周期長度。例如，如果數據是按月份統計的，則可以將 freq 設置為 12，表示每年有 12 個月。
- `type`: 指定平滑方法的類型。通常可以設置為 “`lambda`” 或 “`mu`”，前者適用於數據變化較緩慢的情況，後者適用於數據變化較快的情況。

函數返回一個列表，其中包含這個時間序列的趨勢和週期兩部分，可以通過 `trend` 和 `cycle` 這兩個名稱訪問。趨勢部分是時間序列的長期變化，而週期部分是短期變化，通常被視為噪聲或誤差。通常，趨勢是為了對時間序列進行預測和分析而提取的重要信息。

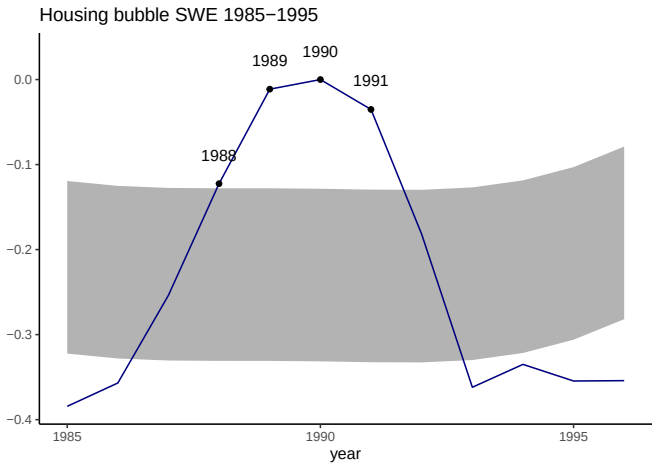
- `year()` 取出每個年份的年份部分，這樣可以只顯示年份而省略月份和日期。
- `nudge_y` 參數用於微調文字標籤的垂直位置，它將文字標籤往上或往下微調一定的距離，以避免與數據點重疊。在這個例子中，`nudge_y = std.cycle/3` 意味著將標籤往上微調一個標準週期的三分之一，這樣可以使標籤位置相對於數據點稍微往上一些，讓它們更加清晰可見。

```
par(mfcol=c(1,3))
get.housing.bubble.plot(housing.price, "Housing bubble", "AUS", 1884, 1894)
```

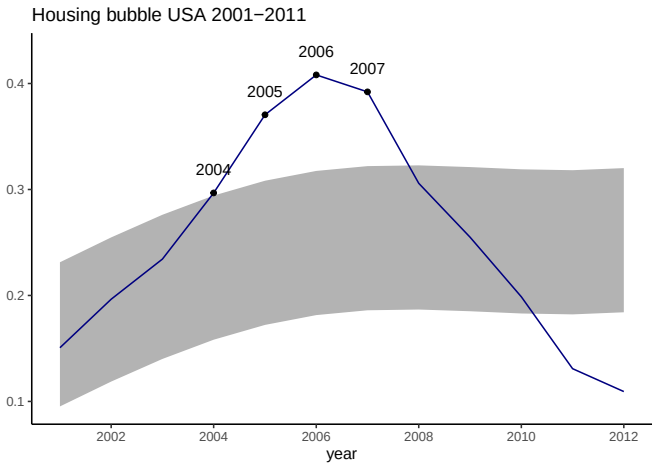
```
## Warning: Using one column matrices in `filter()` was deprecated in dplyr 1.1.0.
## i Please use one dimensional logical vectors instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
get.housing.bubble.plot(housing.price,"Housing bubble", "SWE", 1985, 1995)
```




```
get.housing.bubble.plot(housing.price,"Housing bubble",
                "USA", 2001, 2011)
```



3 Global Crises Data 資料庫

Global Crises Data 資料庫是哈佛大學商學院的行為財務學和金融穩定中心建立的一個資料庫，用於收集全球各國自 20 世紀 70 年代以來的貨幣危機和銀行危機數據。

該 Global Crises Data 資料庫中的 Currency Crises 資料庫的資料來源是經濟學家 Carmen Reinhart 和 Kenneth Rogoff 在 2008 年出版的著作 “This Time Is Different: Eight Centuries of Financial Folly”。該書是對世界各國在過去 800 年間發生的金融危機進行的經濟分析，並且收錄了大量的金融危機相關的數據。Currency Crises 資料庫是該書中的一個章節，收錄了各國在不同時間發生的貨幣危機相關的數據。

本次作業使用的資料來自 Morris Goldstein 和 Mohsin Khan 共同撰寫的 Assessing Financial Vulnerablility，貨幣危機資料庫的資料來源是 International Monetary Fund (IMF) 的資料。該資料庫包含了 1970 年至 2001 年間的貨幣危機事件。所使用的資料庫時間範圍較小，僅包含了 1970 年至 2001 年的資料。

同時也引用了 Kaminsky 和 Reinhart 在 1999 年發表的論文 “The Twin Crises: The Causes of Banking and Balance-of-Payments Problems”，該論文是貨幣危機研究領域中非常重要的一篇論文，以及 Kaminsky, Graciela L., Saul Lizondo, 和 Carmen M. Reinhart 在 1998 年發表的論文 “Leading Indicators of Currency Crises”，該論文提出了一些貨幣危機的預警指標，被廣泛應用於實務中。