EPS贸易数据分析

胡华平

2020-12-25

Table of Contents

# 目前进展

数据收集进展情况：

1. EPS数据库。

* HS2002子库（2002-2011年）：正在下载中，一天内可以完成数据下载！
* HS2012子库（2012-2016年）：已经完成下载！可以进行数据分析。

1. 竹云贸易数据库（2017-2020）。受到数据可系统查询设定的约束，下载工作量较大，需要较长时间。

# HS四位编码农产品数量  
n\_prod <- 197  
# 国家数（包括总值）  
n\_country <- 243  
# 年份数（2017-2020年）  
n\_year <- 4  
# 每次操作需要的平均秒数  
n\_operation <- 5  
  
# 估算全部操作需要的时长为 小时数  
(time\_total <- (n\_prod\*n\_country \*n\_year \*n\_operation)/3600/24)

[1] 11

# 面临问题

# 文献讨论

理论模型：

实证模型：

操作步骤和主要过程

主要结论

# 数据分析

## 比较HS国家编码变动

## 合并HS2012数据

下面合并操作只需要运行一次即可。

all\_files <- list.files(here("data/eps/import/HS2012"))  
n <- length(all\_files)  
  
path\_dir <- here::here("manuscript","data","eps", "import", "HS2012")  
path\_file <- paste0(path\_dir, "/",all\_files)  
  
tbl\_hs2012 <- NULL  
for (i in 1:n) {  
 tbl\_tem <- read\_rds(file = path\_file[i])  
 tbl\_hs2012 <- bind\_rows(tbl\_hs2012, tbl\_tem)  
 print(paste0("成功合并到了第", i,"个文件。共有",n,"个文件！"))  
}  
  
# 写出文件  
write\_rds(tbl\_hs2012, "../data/eps/tbl-hs2012.rds")

## 比较HS产品编码变动

## HS2012数据描述性分析

读取已经合并好的数据，并进行简单清洗，具体包括：

* 去掉原数据中的**千分位分隔符**（逗号,）。
* 正确变换列类型，例如列value（贸易额）的类型应该为number（数值型）。
* 变换数值单位。例如列value（贸易额）的单位由**美元**变换为**万美元**。
* 筛除不必要的数据行。例如，删除所有**贸易量**行（用不到），以及**贸易额**的无数据行（因为没有产生贸易活动）。
* 加入**国家**的编码等信息。需要用到tbl\_nation的数据表，先处理好匹配需要的key，然后使用left\_jion()函数进行匹配。
* 删除重复行，例如产品0909就存在重复情形（原因可能来自于最开始的数据抓取环节）

# 读取数据  
tbl\_hs2012 <- read\_rds("../data/eps/tbl-hs2012.rds")  
  
tbl\_nation <- read\_rds("../data/eps/list-nation-hs2012.rds") %>%  
 rename("country" = "nation", "country\_code"="code",  
 "country\_id"="index")  
#str(tbl\_nation)  
  
# 转换数据形式  
tbl\_hsclear <- tbl\_hs2012 %>%  
 mutate(value = str\_replace\_all(value, ",", "")) %>%  
 mutate(value = as.numeric(value),  
 value = 0.001\*value) %>%  
 filter(str\_detect(var, "美元"), !is.na(value)) %>%  
 left\_join(., tbl\_nation, by = "country")%>%  
 arrange(code\_prod, year,month, country\_id) %>%  
 distinct()  
  
#head(tbl\_hsclear)  
#unique(tbl\_hsclear$year)  
#str(tbl\_hsclear)

下面我们进行初步的数据汇总分析。

1.按产品和按月份，来统计加总进口额情况

这是一张表

表 1: 2012年1001产品分月进口总额

|  |  |  |
| --- | --- | --- |
| code\_prod | month | total |
| 1001 | 05 | 16.76 |
| 1001 | 03 | 16.23 |
| 1001 | 09 | 16.23 |
| 1001 | 10 | 11.09 |
| 1001 | 02 | 10.84 |
| 1001 | 07 | 7.98 |
| 1001 | 04 | 7.65 |
| 1001 | 08 | 6.80 |
| 1001 | 01 | 6.67 |
| 1001 | 06 | 5.62 |
| 1001 | 11 | 4.06 |
| 1001 | 12 | 0.21 |

下面我们来做一张柱状图：

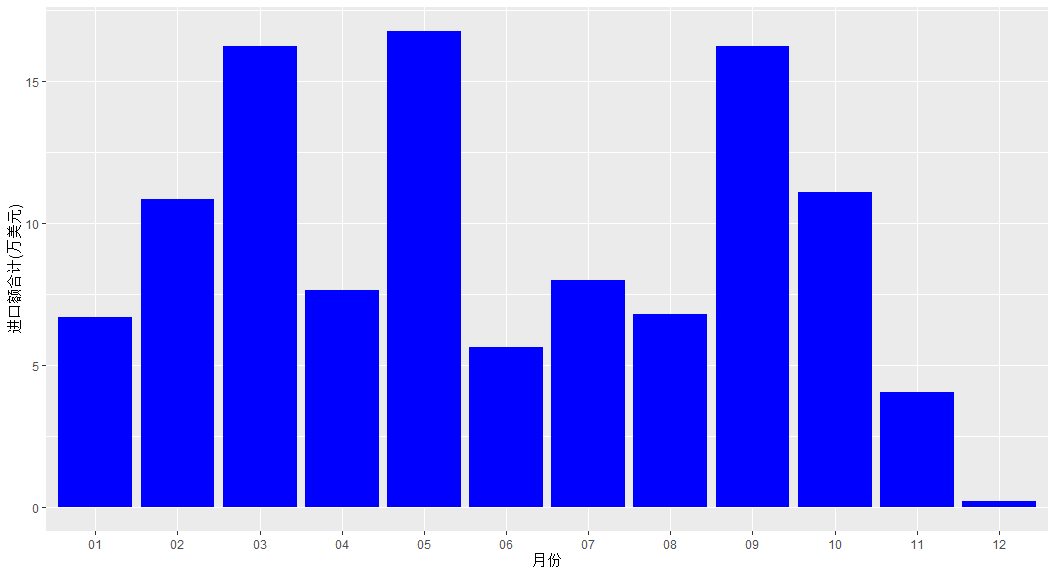


图 1: 一张丑丑的柱状图

上面的表1 和图1 共同表明，分月合计的产品进口贸易额呈现一定的双峰分布形态。

这是另外一张表。它表明什么社么

表 2: 2012年所有产品分月进口总额

|  |  |
| --- | --- |
| month | sum\_pro |
| 09 | 7908597 |
| 07 | 7659008 |
| 12 | 7571235 |
| 05 | 7395318 |
| 06 | 7237374 |
| 08 | 7200955 |
| 11 | 6929048 |
| 03 | 6893233 |
| 04 | 6779086 |
| 10 | 6713241 |
| 01 | 5988275 |
| 02 | 5665867 |

分析国家进出情况（有问题，未完成）。

tbl\_smry <- tbl\_hsclear %>%  
 ungroup() %>%  
 filter(year=="2012") %>%  
 filter(country !="总值") %>%  
 group\_by(code\_prod, month) %>%  
 summarise(n = dplyr::n(country))

# 得到分析数据表

## 贸易总额及其月度变动

### 理论表达

表示**分产品每个月**的加总贸易额，表示滞后1期变量。

表示**分产品**下的**月度贸易变动额**。

### 代码操作

**具体思路**如下：

1. 我们注意到原数据中实际上已经有**分产品**下的**月度贸易总额**。数据。也即country变量下包含有总值类别，因此可以直接使用filter()函数过滤得到**分产品**下的**月度贸易总额**。
2. **分产品**下的**月度贸易变动额**可以先计算得到滞后变量，然后再差分得到贸易变动额（$\Delta X}= X\_{l0} - X\_{l1}$）。具体使用group\_by() + dplyr::lag()函数操作得到滞后变量。

以下为具体R代码操作：

xc <- tbl\_hsclear %>%  
 arrange(code\_prod, year, month) %>%  
 filter(country\_id==1) %>%  
 #unite(col = "ym", year, month, sep = "-", remove = FALSE) %>%  
 select(code\_prod,product, year, month, value) %>%  
 group\_by(product) %>%  
 mutate(x\_l0 = value,   
 x\_l1 = dplyr::lag(value,n=1)) %>%  
 ungroup() %>%  
 mutate(delta\_x = x\_l0 - x\_l1)

### 过程解释

下面表3简单展示了得到的数据表结果：

表 3: （其中5类产品）分产品的月度贸易额及变动（单位：万美元）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| index | code\_prod | product | year | month | x\_l0 | x\_l1 | delta\_x |
| 344 | 0201 | 0201 - 鲜、冷牛肉 | 2016 | 09 | 5772 | 4559 | 1213 |
| 345 | 0201 | 0201 - 鲜、冷牛肉 | 2016 | 10 | 5375 | 5772 | -397 |
| 346 | 0201 | 0201 - 鲜、冷牛肉 | 2016 | 11 | 5569 | 5375 | 194 |
| 347 | 0201 | 0201 - 鲜、冷牛肉 | 2016 | 12 | 7262 | 5569 | 1694 |
| 4323 | 1001 | 1001 - 小麦及混合麦 | 2016 | 09 | 102137 | 87307 | 14830 |
| 4324 | 1001 | 1001 - 小麦及混合麦 | 2016 | 10 | 60288 | 102137 | -41850 |
| 4325 | 1001 | 1001 - 小麦及混合麦 | 2016 | 11 | 6282 | 60288 | -54006 |
| 4326 | 1001 | 1001 - 小麦及混合麦 | 2016 | 12 | 47432 | 6282 | 41150 |
| 4509 | 1005 | 1005 - 玉米 | 2016 | 09 | 5160 | 6933 | -1773 |
| 4510 | 1005 | 1005 - 玉米 | 2016 | 10 | 5303 | 5160 | 143 |
| 4511 | 1005 | 1005 - 玉米 | 2016 | 11 | 11522 | 5303 | 6219 |
| 4512 | 1005 | 1005 - 玉米 | 2016 | 12 | 38729 | 11522 | 27207 |
| 4569 | 1006 | 1006 - 稻谷、大米 | 2016 | 09 | 101107 | 87018 | 14088 |
| 4570 | 1006 | 1006 - 稻谷、大米 | 2016 | 10 | 95166 | 101107 | -5941 |
| 4571 | 1006 | 1006 - 稻谷、大米 | 2016 | 11 | 142753 | 95166 | 47587 |
| 4572 | 1006 | 1006 - 稻谷、大米 | 2016 | 12 | 199596 | 142753 | 56843 |
| 5256 | 1201 | 1201 - 大豆，不论是否破碎 | 2016 | 09 | 3078517 | 3292747 | -214230 |
| 5257 | 1201 | 1201 - 大豆，不论是否破碎 | 2016 | 10 | 2248721 | 3078517 | -829795 |
| 5258 | 1201 | 1201 - 大豆，不论是否破碎 | 2016 | 11 | 3382301 | 2248721 | 1133579 |
| 5259 | 1201 | 1201 - 大豆，不论是否破碎 | 2016 | 12 | 3927236 | 3382301 | 544936 |

## 贸易变动分解

### 理论表达

其中：

* 表示由于新国家加入（Newcoming）贸易（与上月相比），引发的贸易额变动:
* 表示由于原国家退出（Exit）贸易（与上月相比），引发的贸易额变动：
* 表示由于**继续留存**国家**贸易量**增加（Growth）（与上月相比），引发的贸易额变动：
* 表示由于**继续留存**国家**贸易量**缩减（Shrink）（与上月相比），引发的贸易额变动：

至此，我们还可以分别计算上述4种情形下的：贸易变动()、国家名单()、国家数量()、贸易密度()，具体变量定义为：

1. **新进入**情形(Newcoming)：贸易变动()、国家名单、国家数量，贸易密度。
2. **退出**情形(Exit)：贸易变动()、国家名单、国家数量，贸易密度。
3. 继续留存下业务量**增长**情形(Growth)：贸易变动()、国家名单、国家数量，贸易密度。
4. 继续留存下业务量**缩减**情形(Shrink)：贸易变动()、国家名单、国家数量，贸易密度。

### 代码操作

**基本思路**：

* 实现新的数据形式变换（按产品分年月，“折叠”行数据）。利用group\_by() + nest()函数，构造出list-column data.frame。
* 构造**当期**的数据块和**滞后期**的数据块。具体要用到lag()函数。

要注意对初始项的处理，可以是“保持”、“NA”。我们选择了“保持”。

* 比较**当期**数据块和**滞后期**数据块的差异，分别计算得到**进入或退出**的国家名单、国家数量、贸易密度、贸易额变动。

具体要用到平行计算包purrr::map2()函数，同时还需要编写专门的函数compare.tbls()函数来得到自己想要的上述几个变量。

具体的R代码如下：

注意：因为数据集比较大，而且设计到多个计算，尽管purrr::map2()已经进行了平行化运算，但是还是会耗费一定计算时间，大约1-2分钟。因此，**强烈建议**一次计算，然后保留结果。

首先，编写定制函数compare.tbls()：

#dt2 <- tbl\_compare$dt\_l0[1][[1]]  
#dt1 <- tbl\_compare$dt\_l1[1][[1]] %>%  
 #mutate(country=c(LETTERS[1:5]))  
  
# build function for map2   
## it can compare difference with two data.frame, also bypass some calculate  
compare.tbls <- function(dt1, dt2){  
 # new coming country  
 c\_new <- setdiff(dt2$country, dt1$country)  
 n\_new <- length(c\_new)  
 v\_new <- dt2 %>%  
 filter(country %in% c\_new) %>%  
 summarise(value = sum(value, na.rm = T)) %>%  
 unlist()  
 d\_new <- ifelse(n\_new!=0, v\_new/n\_new, NA)  
   
 # exit country  
 c\_exit <- setdiff(dt1$country,dt2$country )  
 n\_exit <- length(c\_exit)  
 v\_exit <- dt1 %>%  
 filter(country %in% c\_exit) %>%  
 summarise(value = sum(value, na.rm = T)) %>%  
 unlist()  
 d\_exit <- ifelse(n\_exit!=0, v\_exit/n\_exit, NA)  
   
 # comon country  
 common\_c <- intersect(dt2$country, dt1$country)  
 dt2\_com <- dt2 %>%  
 filter(country %in% common\_c) %>%  
 add\_column(period = "T2", .before = "country")  
 dt1\_com <- dt1 %>%  
 filter(country %in% common\_c)%>%  
 add\_column(period = "T1", .before = "country")  
 ### case if no common country  
 dt\_null <- data.frame(matrix(rep(NA,6),nrow = 1 ))   
 names(dt\_null) <- c('country','country\_id','T1','T2','status','change')  
 #paste0(paste0("'",names(dt\_com) ,"'"), collapse = ",")  
 if (length(common\_c)==0) {  
 dt\_com <- dt\_null %>%  
 filter(!is.na(country))  
 } else{  
 dt\_com <- bind\_rows(dt2\_com, dt1\_com) %>%  
 spread(key = period, value = value) %>%  
 mutate(status = ifelse(T2<T1, "S", "G"),  
 change = T2-T1)  
 }  
   
 c\_com <- unlist(dt\_com$country)  
 n\_com <- length(c\_com)   
   
 # for grow country  
 dt2\_grow <- dt\_com %>%  
 filter(status=="G")  
 c\_grow <- unlist(dt2\_grow$country)  
 n\_grow <- length(c\_grow)  
 v\_grow <- dt2\_grow %>%  
 summarise(value = sum(change, na.rm = T)) %>%  
 unlist()  
 d\_grow <- ifelse(n\_grow!=0, v\_grow/n\_grow, NA)  
   
 # for shrink country  
 dt2\_shrink <- dt\_com %>%  
 filter(status=="S")  
 c\_shrink <- unlist(dt2\_shrink$country)  
 n\_shrink <- length(c\_shrink)  
 v\_shrink <- dt2\_shrink %>%  
 summarise(value = sum(change, na.rm = T)) %>%  
 unlist()  
 d\_shrink <- ifelse(n\_shrink!=0, v\_shrink/n\_shrink, NA)  
   
 # all results  
 out<- data.frame(c\_new = I(list(c\_new)), n\_new = n\_new,   
 v\_new =v\_new, d\_new = d\_new,  
 c\_exit = I(list(c\_exit)), n\_exit = n\_exit,   
 v\_exit =v\_exit, d\_exit = d\_exit,  
 c\_com = I(list(c\_com)), n\_com = n\_com,  
 c\_grow = I(list(c\_grow)), n\_grow = n\_grow,   
 v\_grow =v\_grow, d\_grow = d\_grow,  
 c\_shrink = I(list(c\_shrink)), n\_shrink = n\_shrink,   
 v\_shrink =v\_shrink, d\_shrink = d\_shrink)  
 return(out)  
}

其次，其余的数据**变换**、**计算**和**导出**过程如下：

# nest dataset and then compare difference  
require(purrr)  
rm\_list <- c("var", "ym", "country\_code")  
tbl\_compare <- tbl\_hsclear %>%  
 select(-all\_of(rm\_list)) %>%  
 filter(country\_id!=1) %>%  
 group\_by(code\_prod, product, year, month) %>%  
 nest() %>%  
 ungroup() %>%  
 rename("dt\_l0"="data") %>%  
 group\_by(code\_prod) %>%  
 mutate(dt\_l1 = lag(dt\_l0,  
 default = list(first(dt\_l0)))) %>%  
 # purrr with custom function compare.tbls  
 mutate(compare = map2(.x = dt\_l1, .y = dt\_l0, .f = compare.tbls))  
  
write\_rds(tbl\_compare, here("data/eps/tbl-compare-newcoming-exit-01-raw.rds"))   
  
# unnest data table  
tbl\_ne <- tbl\_compare %>%  
 select(-dt\_l0, -dt\_l1) %>%  
 unnest(cols = compare)  
  
write\_rds(tbl\_ne, here("data/eps/tbl-compare-newcoming-exit-02-unnest.rds"))

### 过程解释

#### 最开始的数据集

首先，下表4给出了最开始清洗好的数据集tbl\_hsclear，为了演示方便，我们选择**HS编码**为1005的农产品（玉米），年月范围为2012/01-2020/03

表 4: 2012/01-2020/03期间玉米产品月度贸易数据（单位：万美元）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| code\_prod | product | year | month | country | country\_id | value |
| 1005 | 1005 - 玉米 | 2012 | 01 | 印度 | 12 | 7.64 |
| 1005 | 1005 - 玉米 | 2012 | 01 | 老挝 | 20 | 196.79 |
| 1005 | 1005 - 玉米 | 2012 | 01 | 德国 | 108 | 1702.94 |
| 1005 | 1005 - 玉米 | 2012 | 01 | 智利 | 161 | 0.42 |
| 1005 | 1005 - 玉米 | 2012 | 01 | 秘鲁 | 182 | 74.47 |
| 1005 | 1005 - 玉米 | 2012 | 01 | 美国 | 197 | 248619.63 |
| 1005 | 1005 - 玉米 | 2012 | 02 | 印度 | 12 | 270.00 |
| 1005 | 1005 - 玉米 | 2012 | 02 | 老挝 | 20 | 1205.00 |
| 1005 | 1005 - 玉米 | 2012 | 02 | 德国 | 108 | 1626.48 |
| 1005 | 1005 - 玉米 | 2012 | 02 | 法国 | 109 | 6.49 |
| 1005 | 1005 - 玉米 | 2012 | 02 | 奥地利 | 119 | 1.29 |
| 1005 | 1005 - 玉米 | 2012 | 02 | 阿根廷 | 153 | 12.95 |
| 1005 | 1005 - 玉米 | 2012 | 02 | 巴西 | 159 | 3.93 |
| 1005 | 1005 - 玉米 | 2012 | 02 | 秘鲁 | 182 | 42.13 |
| 1005 | 1005 - 玉米 | 2012 | 02 | 美国 | 197 | 171087.71 |
| 1005 | 1005 - 玉米 | 2012 | 02 | 澳大利亚 | 200 | 54.00 |
| 1005 | 1005 - 玉米 | 2012 | 03 | 缅甸 | 7 | 51.26 |
| 1005 | 1005 - 玉米 | 2012 | 03 | 老挝 | 20 | 420.38 |
| 1005 | 1005 - 玉米 | 2012 | 03 | 德国 | 108 | 962.08 |
| 1005 | 1005 - 玉米 | 2012 | 03 | 法国 | 109 | 114.50 |
| 1005 | 1005 - 玉米 | 2012 | 03 | 俄罗斯联邦 | 142 | 21.00 |
| 1005 | 1005 - 玉米 | 2012 | 03 | 阿根廷 | 153 | 12.21 |
| 1005 | 1005 - 玉米 | 2012 | 03 | 智利 | 161 | 0.69 |
| 1005 | 1005 - 玉米 | 2012 | 03 | 圭亚那 | 173 | 1.58 |
| 1005 | 1005 - 玉米 | 2012 | 03 | 美国 | 197 | 151960.54 |

#### 分年月按产品“折叠”后的数据集

简要信息如下：

tbl\_compare <- read\_rds(here("data/eps/tbl-compare-newcoming-exit-01-raw.rds"))  
  
#dplyr::glimpse(tbl\_compare)

分年月按产品“折叠”后的数据集，详细的数据集形态见图2

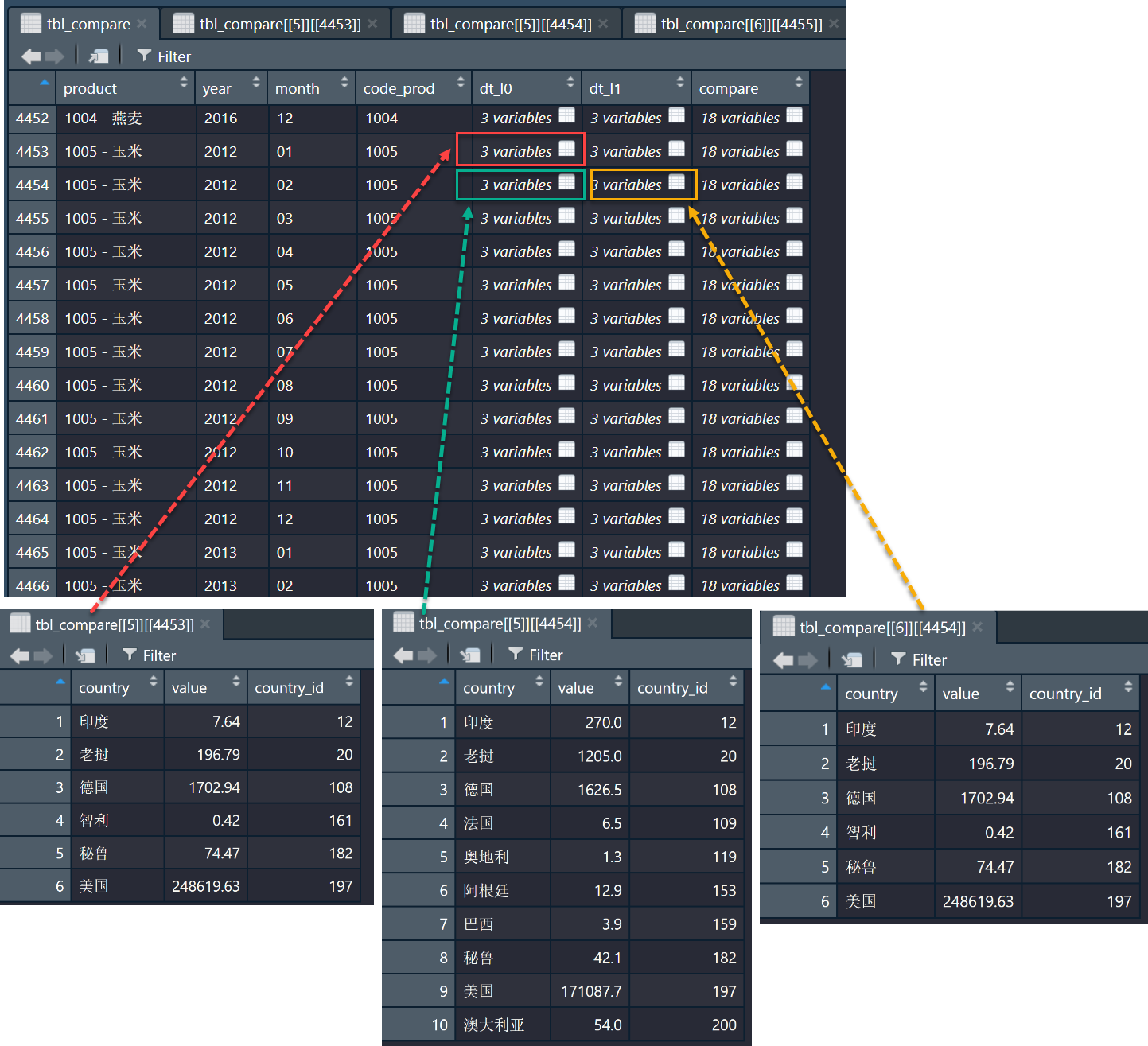


图 2: 步骤2：数据“折叠”及形态关系

#### 进行月度比较得到的变动分解数据集

首先，我们需要：

* 读取数据文件.rds
* 去掉分组属性ungroup()
* 将0全部替换为NA。具体需要用到dplyr::na\_if()函数

下表5中，我们将看到对年月贸易变动的计算和比较数据集示例：

表 5: 步骤3：贸易变动计算

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| product | year | month | code\_prod | n\_new | v\_new | d\_new | n\_exit | v\_exit | d\_exit | n\_com | n\_grow | v\_grow | d\_grow | n\_shrink | v\_shrink | d\_shrink |
| 1005 - 玉米 | 2012 | 01 | 1005 |  |  |  |  |  |  | 6 | 6 |  |  |  |  |  |
| 1005 - 玉米 | 2012 | 02 | 1005 | 5 | 79 | 16 | 1 | 0.42 | 0.42 | 5 | 2 | 1271 | 635 | 3 | -77641 | -25880 |
| 1005 - 玉米 | 2012 | 03 | 1005 | 4 | 75 | 19 | 5 | 371.34 | 74.27 | 5 | 1 | 108 | 108 | 4 | -20577 | -5144 |