

```
In [ ]: library(magrittr)
library(dplyr)
library(countrycode)
```

Dataset BACI

Importo lista di file che riportano gli scambi commerciali tra paese e paese, per ogni anno, e un file che contiene il match tra nome del paese e i relativi codici (di varia natura).

```
In [2]: #importo la lista di dataset e il file con il match tra country code e codice ISO3
list_of_files <- list.files(path = "C:dataset/", recursive = TRUE,
                             pattern = "\\.",
                             full.names = TRUE)
head(read.table(list_of_files[1], sep=',', header=TRUE))
```

A data.frame: 6 × 6

	t	i	j	k	v	q
	<int>	<int>	<int>	<int>	<dbl>	<chr>
1	1996	4	12	90930	226.647	170.648
2	1996	4	12	130190	17.720	14.000
3	1996	4	20	570110	25.003	1.625
4	1996	4	32	320611	1.049	0.016
5	1996	4	32	392490	6.031	2.375
6	1996	4	32	570110	20.994	0.789

```
In [3]: idx_code <- read.csv("country_codes_BACI.csv",
                             sep=",", header=TRUE)
head(idx_code)
```

A data.frame: 6 × 4

	country_code	country_name	country_iso2	country_iso3
	<int>	<chr>	<chr>	<chr>
1	4	Afghanistan	AF	AFG
2	8	Albania	AL	ALB
3	12	Algeria	DZ	DZA
4	16	American Samoa	AS	ASM
5	20	Andorra	AD	AND
6	24	Angola	AO	AGO

Verifico che la trasformazione in ISO3c corrisponda alla stessa trasformazione di funzione countrycode in R, in modo che siano coerenti con la denominazione dei paesi negli altri dataset. Alcuni li sostituisco manualmente (i nomi non sono scritti in modo corretto, il sistema non li riconosce), altri lascio un ISO3c uguale a NA perchè corrispondono a entità geografiche troppo generiche o non più esistenti con quel nome.

```
In [4]: paesi_iso3c <- countrycode(idx_code$country_name, origin = "country.name",
                                   destination = "iso3c")
paesi <- as.data.frame(cbind(name = idx_code$country_name, paesi_iso3c))
```

Warning message:

"Some values were not matched unambiguously: Belgium-Luxembourg (...1998), Bonaire, Br. Indian Ocean Terr., Br. Virgin Isds, Curaçao, Czechoslovakia (...1992), Europe EFTA, nes, Fr. South Antarctic Terr., FS Micronesia, Mayotte (Overseas France), Netherlands Antilles (...2010), Other Asia, nes, Saint Barthélemy, Serbia and Montenegro (...2005), Southern African Customs Union (...1999), TÅrkiye
"

Warning message:

"Some strings were matched more than once, and therefore set to <NA> in the result: Mayotte (Overseas France), FRA, MYT
"

```
In [5]: paesi_iso3c[28] <- "VGB"
paesi_iso3c[48] <- "MYT"
paesi_iso3c[144] <- "CUW"
paesi_iso3c[147] <- "BES"
paesi_iso3c[158] <- "FSM"
paesi_iso3c[176] <- "BLM"
paesi_iso3c[217] <- "TUR"
paesi_iso3c[224] <- NA #urss non la conto

#sostituisco questi nella colonna idx_code
idx_code$country_iso3c <- paesi_iso3c
```

Unisco i diversi dataset per ogni anno in unico dataset formato panel, eliminando le colonne che riportano i beni a livelli disaggregato e la quantità di beni espressa in kg. Infine, aggrego le righe che hanno lo stessa *origin*, *destination* e *year* sommando il valore in USD dei beni.

```
In [6]: full_data<-NULL
baci <- NULL

for (i in 1:length(list_of_files)){
  dat<-read.table(list_of_files[i],sep=',', header=TRUE)
  full_data<- dat

  new_data<-full_data %>%
    select (-c(q)) #questa è la quantità in kg

  colnames(new_data)<-c("year","exporter","importer","product","value_baci")

  new_new<-new_data %>% #qua perchè non interessa il tipo di prodotto
    select (-c(product))
```

```

suppressMessages(
  data_merge <- new_new %>% #quindi aggreghi
    group_by(year,exporter,importer) %>%
    summarise(value_baci = sum(value_baci))
)
full_baci<-data_merge
colnames(full_baci)<-c("year","origin","destination","value_baci")
full_baci <- full_baci %>%
  select(origin,destination,year,value_baci)

full_baci$value_baci <-full_baci$value_baci*1000
  #perchè sono espressi in "migliaia di USD"
baci <- rbind(baci, full_baci)
}

```

Sostituisco il codice ISO3n usato nel dataset BACI con il codice ISO3c corrispondente determinato come riportato sopra.

```

In [7]: baci$origin <- idx_code$country_iso3c[
          match(as.factor(baci$origin),
                as.factor(idx_code$country_code))]
baci$destination <- idx_code$country_iso3c[
          match(as.factor(baci$destination),
                as.factor(idx_code$country_code))]
baci <- na.omit(baci)

```

Creo un dataset con tutte le possibili combinazioni tra *origin*, *destination* e *year* (tra quelli presenti nel dataset originale).

```

In [8]: p <- as.character(na.omit(paesi_iso3c))
y <- unique(baci$year)
tot <- as.data.frame(cbind(rep(p, each = length(p)*length(y)),
                           rep(rep(p,times=length(p)),each=length(y)),
                           rep(y, times = length(p)^2)))
colnames(tot) <- c("origin","destination","year")
tot <- tot[-c(which(tot$origin ==
                    tot$destination)),]
tot <- cbind(tot,rep(0,nrow(tot)))
colnames(tot) <- c("origin","destination","year","value_baci")
tot$year <-
  as.numeric(tot$year)
tot$value_baci <-
  as.numeric(tot$value_baci)
merge <- rbind(baci,tot)
baci_tot <- merge %>%
  group_by(origin,destination,year) %>%
  summarise(value_baci = sum(value_baci))

```

``summarise()`` has grouped output by 'origin', 'destination'. You can override using the ``groups`` argument.

```

In [9]: baci_tot <- baci_tot[order(baci_tot$year),]
write.csv(baci_tot,"baci_tot.csv", row.names = FALSE)

```