In [ ]:

```r
library(magrittr)
library(dplyr)
library("countrycode")
```

Dataset USICT

Importo lista di dataset, divisi per anno.

In [10]:

```r
list_of_files <- list.files(path = "C:dataset/", recursive = TRUE,
                            pattern = "\\.",
                            full.names = TRUE)
```

In [11]:

```r
dat <- read.csv(list_of_files[1],sep=',', header=TRUE)
head(dat[,1:4])
head(dat[,5:9])
head(dat[,10:13])
```

A data.frame: 6 × 4

	exporter_iso3	exporter_dynamic_code	exporter_name	importer_iso3
	<chr>	<chr>	<chr>	<chr>
1	AUS	AUS	Australia	SGP
2	AUS	AUS	Australia	USA
3	AUT	AUT	Austria	DEU
4	AUT	AUT	Austria	HUN
5	AUT	AUT	Austria	LTU
6	AUT	AUT	Austria	POL

A data.frame: 6 × 5

	importer_dynamic_code	importer_name	broad_sector	industry_id	industry_descr
	<chr>	<chr>	<chr>	<int>	<chr>
1	SGP	Singapore	Services	154	Manufacturing services on physical inputs owned by others
2	USA	United States	Services	154	Manufacturing services on physical inputs owned by others
3	DEU	Germany	Services	154	Manufacturing services on physical inputs owned by others
4	HUN	Hungary	Services	154	Manufacturing services on physical inputs owned by others
5	LTU	Lithuania	Services	154	Manufacturing services on physical inputs owned by others
6	POL	Poland	Services	154	Manufacturing services on physical inputs owned by others

A data.frame: 6 × 4

	year	trade	flag_mirror	flag_zero
	<int>	<dbl>	<int>	<chr>
1	2005	0.000	0	r
2	2005	4.600	0	p
3	2005	154.639	1	p
4	2005	9.032	1	p
5	2005	0.126	1	p
6	2005	1.114	1	p

Creo dataset unico e mantengo la colonna che riporta il nome del paese origine dove risiede il creditore (exporter_name), la colonna che riposta il nome del paese di destinazione dove risiede il debitore (importer_name), la colonna dell'anno e del valore dello scambio finanziario (trade).

In [9]:
```
full_data<-NULL
usict <- NULL
for (i in 1:length(list_of_files)){
  dat<-read.csv(list_of_files[i],sep=',', header=TRUE)
  full_data <- rbind(full_data,dat)
}
```

In [20]:
```
new_data<- full_data %>%
  select (-c(exporter_dynamic_code, exporter_iso3,importer_dynamic_code,
             importer_iso3,broad_sector,industry_id,
             industry_descr,flag_mirror,flag_zero))
usict <- new_data %>%
  group_by(exporter_name,importer_name,year) %>%
  summarise(trade = sum(trade))
```

```
`summarise()` has grouped output by 'exporter_name', 'importer_name'. You can overri
de using the `.groups` argument.
```

In [21]:
```
#ci sono dati in cui origin = destination
usict <- usict[-c(which(usict$exporter_name == usict$importer_name)),]
colnames(usict)<- c("origin","destination","year","value_usics")
head(usict)
```

A grouped_df: 6 × 4

origin	destination	year	value_usics
<chr>	<chr>	<int>	<dbl>
Afghanistan	Belarus	2004	0.0000
Afghanistan	Belarus	2005	0.0000
Afghanistan	Belarus	2006	0.0068
Afghanistan	Belarus	2007	0.0018
Afghanistan	Belarus	2008	0.0016
Afghanistan	Belarus	2009	0.0060

In [28]:
```
paesi_nomi <- c(unique(usict$origin),setdiff(
    unique(usict$destination),unique(usict$origin)))
paesi_iso3c <- countrycode(paesi_dataset,origin = "country.name",destination ="iso3
paesi <- as.data.frame(cbind(paesi_nomi,paesi_iso3c))
paesi$paesi_iso3c[63] <- "ETH"
```

```
Warning message:
"Some values were not matched unambiguously: Ethiopia (excludes Eritrea)
"
Warning message:
"Some strings were matched more than once, and therefore set to <NA> in the result:
Ethiopia (excludes Eritrea),ERI,ETH
"
```

Sostituisco il nome del paese con il corrispettivo codice ISO3c.

In [29]:
```
usict$origin <- paesi$paesi_iso3c[match(as.factor(usict$origin),
                                                  as.factor(paesi$paesi_nomi))]
usict$destination <- paesi$paesi_iso3c[match(as.factor(usict$destination),
                                                       as.factor(paesi$paesi_nomi))
```

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

In [34]:
```
p <- paesi$paesi_iso3c
y <- unique(usict$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_usict")
tot$year <- as.numeric(tot$year)
merge <- rbind(usict,tot)
usict_tot <- merge %>%
  group_by(origin,destination,year) %>%
  summarise(value_usict = sum(value_usict))
```

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

In [37]:
```
usict_tot <- usict_tot[order(usict_tot$year),]
write.csv(usict_tot,"usict_tot.csv", row.names = FALSE)
```