

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

## TOURISM OUTBOUND - UNWTO

Importo la lista di file contenenti i dati relativi al flussi di turisti: ogni file è denominato con il nome del paese di partenza e al suo interno c'è un dataframe le cui righe corrispondono al paese di destinazione e le cui colonne indicano l'anno in cui si è registrato il dato sui turisti. I turisti, dal paese di destinazione, possono essere classificati nei seguenti modi:

- TFN : Arrivi di turisti non residenti alle frontiere nazionali, registrati per nazionalità.
- TFR : Arrivi di turisti non residenti alle frontiere nazionali, registrati per Paese di residenza
- VFN : Arrivi di visitatori non residenti alle frontiere nazionali, registrati per nazionalità
- VFR : Arrivi di visitatori non residenti alle frontiere nazionali, registrati per Paese di residenza
- THSN : Arrivi di turisti non residenti in alberghi e strutture simili, registrati per nazionalità
- THSR : Arrivi di turisti non residenti in alberghi e strutture simili, registrati per paese di residenza.
- TCEN : Arrivi di turisti non residenti in tutti i tipi di strutture ricettive, registrati per nazionalità
- TCER : Arrivi di turisti non residenti in tutti i tipi di esercizi ricettivi, registrati per paese di residenza.

Alcuni paesi registrano le persone in entrata con solo una di queste classificazioni, altri con più di una.

```
In [2]: list_of_files <- list.files(path = "C:dataset/", recursive = TRUE,
                                pattern = "\\.",
                                full.names = TRUE)
cut <- sub("C:dataset/", "", list_of_files)
paesi_nomi <- sub(" OUTBOUND ALL SERIES 1995-2021.xlsx", "", cut)
#così ho recuperato i paesi
```

Converto i nomi dei paesi con il loro rispettivo ISO3c. La funzione countrycode non attribuisce a Bonaire, Saba e Sint Eustatius un ISO3c perchè è lo stesso per tutti e tre quindi lo inseriamo manualmente.

```
In [3]: paesi_iso3c <- countrycode(paesi_nomi, origin = "country.name",
                                destination = "iso3c")
```

```
#ne mancano solo tre che hanno lo stesso iso3c: BES, sostituisco manualmente  
paesi_iso3c[which(is.na(paesi_iso3c))] <- "BES"
```

Warning message:

```
"Some values were not matched unambiguously: Bonaire, Saba, Sint Eustatius  
"
```

Uniamo i diversi dataset per ogni paese di partenza in un unico dataset. Rimuovo le prime sei righe che contengono informazioni non utili e l'ultima colonna che contiene la variazione degli arrivi tra il 2020 e il 2021. Uso la funzione melt per riscrivere il dataset in formato panel.

```
In [4]: outbound_tot <- NULL  
pb <- txtProgressBar(min = 0, max = length(list_of_files), initial = 0)  
for(i in 1:length(list_of_files)){  
  setTxtProgressBar(pb,i)  
  data <- suppressMessages(read_excel(list_of_files[i]))  
  new_data <- as.data.frame(data[-c(1:6),-30])  
  colnames(new_data) <- c("destination", "SERIES", c(1995:2021))  
  new_data <- new_data[-c(which(is.na(new_data$destination))),]  
  #in alcuni dataset vengono importate delle righe con soli NA  
  new_data[is.na(new_data)] <- 0  
  resh_data <- melt(new_data, id=c(1,2))  
  nomi_1 <- unique(resh_data$destination)  
  code_1 <- countrycode(nomi_1, origin = "country.name", destination = "iso3c")  
  paesi_1 <- as.data.frame(cbind(nomi_1, code_1))  
  resh_data$destination <- paesi_1$code_1[match(as.factor(resh_data$destination),  
                                              as.factor(paesi_1$nomi_1))]  
  
  data_series <- resh_data %>%  
  mutate(value_vfn = ifelse(SERIES == "VFN", value, 0),  
         value_tfr = ifelse(SERIES == "TFR", value, 0),  
         value_vfr = ifelse(SERIES == "VFR", value, 0),  
         value_tfn = ifelse(SERIES == "TFN", value, 0),  
         value_tcer = ifelse(SERIES == "TCER", value, 0),  
         value_thsr = ifelse(SERIES == "THSR", value, 0),  
         value_tcen = ifelse(SERIES == "TCEN", value, 0),  
         value_thsn = ifelse(SERIES == "THSN", value, 0))  
  #la colonna delle classificazioni e la colonna dei valori non distribuiti secondo L  
  outbound <- data_series[, -c(2,4)]  
  outbound <- as.data.frame(cbind(rep(paesi_iso3c[i], times=nrow(outbound)), outbound,  
                                colnames(outbound)[1] <- "origin"  
                                outbound_tot <- rbind(outbound_tot, outbound)  
  }  
}
```

=====

```
In [5]: colnames(outbound_tot)[3] <- "year"  
for(i in 4:11){outbound_tot[,i] <- as.numeric(outbound_tot[,i])}  
#mi dice NA introduces by coercion  
#ci sono numeri scritti male già nel file excel  
  
names <- colnames(outbound_tot)[4:11]  
outbound_fin <- outbound_tot %>%  
  group_by(origin, destination, year) %>%  
  summarise(across(all_of(names), sum))  
outbound_fin <- outbound_fin[order(outbound_fin$year),]
```

```
outbound_fin <- outbound_fin[-c(which(outbound_fin$origin ==  
                                     outbound_fin$destination)),]
```

Warning message in eval(expr, envir, enclos):

"NA introdotti per coercizione"

Warning message in eval(expr, envir, enclos):

"NA introdotti per coercizione"

Warning message in eval(expr, envir, enclos):

"NA introdotti per coercizione"

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

Creo un dataset in cui ci siano tutte le possibili combinazioni tra paese di origine, di destinazione e anno (tra quelli già presenti nel dataset).

```
In [6]: p <- unique(paesi_iso3c)  
y <- levels(unique(outbound_fin$year))  
tot <- as.data.frame(cbind(rep(p, each = length(p)*length(y)),  
                           rep(rep(p,times=length(p)),each=length(y)),  
                           rep(y, times = length(p)^2)))  
colnames(tot) <- c("origin","destination","year")  
tot <- tot[-c(which(tot$origin == tot$destination)),]  
tot <- cbind(tot,replicate(8,rep(0,nrow(tot))))  
colnames(tot) <- colnames(outbound_fin)  
merge <- rbind(outbound_fin,tot)  
outbound_tot_fin <- merge %>%  
  group_by(origin,destination,year) %>%  
  summarise(across(all_of(names),sum))
```

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

```
In [7]: outbound_tot_fin <- outbound_tot_fin[order(outbound_tot_fin$year),]  
write.csv(outbound_tot_fin,"unwto.csv", row.names = FALSE)
```