

# Shiny Dashboard ‘Taux d’inflation’

MADANI Riad

2022-12-14

## Taux d’inflation dans le monde [1980-2022]

Le Groupe des perspectives de la Banque mondiale a construit un jeu de données mondial sur l’inflation. Ce jeu de données couvre jusqu’à 167 pays et plusieurs régions économiques dans le monde sur la période de 1980 au 2022, et comprend la mesure de l’inflation en fréquences annuelle : Inflation globale de l’indice des prix à la consommation (IPC) Nous avons utilisé ce jeu de données pour élaborer un tableau de bord qui nous permet de visualiser le taux d’inflation durant cette période d’années

## Projet Shiny Dashboard

Notre projet contient 3 fichier .R (global, ui et server) et un fichier style.css pour la mise en forme des couleurs et des polices sur notre application

### global.R

Un fichier qui contient toutes les bibliothèques nécessaires a la bonne exécution de notre programme ainsi que l’importation des fichier à utilisés dont le jeu de données et le fichier geojson (“worldcountry” pour les données spatial) Des pré-traitement des données pour une utilisation plus adéquate a nos graphs de visualisation

**Imporation des différents packages si ses derniers ne sont pas installées sur la machine**

```
if(!require(shiny)) install.packages("shiny", repos = "http://cran.us.r-project.org")
if(!require(shinyWidgets)) install.packages("shinyWidgets",
                                             repos="http://cran.us.r-project.org")
if(!require(shinydashboard)) install.packages("shinydashboard",
                                              repos = "http://cran.us.r-project.org")
if(!require(shinythemes)) install.packages("shinythemes",
                                           repos = "http://cran.us.r-project.org")
if(!require(shinycssloaders)) install.packages("shinycssloaders",
                                              repos = "http://cran.us.r-project.org")
if(!require(magrittr)) install.packages("magrittr",
                                       repos = "http://cran.us.r-project.org")
if(!require(rvest)) install.packages("rvest",
                                     repos = "http://cran.us.r-project.org")
```

```

if(!require(readxl)) install.packages("readxl",
                                     repos = "http://cran.us.r-project.org")
if(!require(dplyr)) install.packages("dplyr",
                                     repos = "http://cran.us.r-project.org")
if(!require(maps)) install.packages("maps",
                                     repos = "http://cran.us.r-project.org")
if(!require(ggplot2)) install.packages("ggplot2",
                                     repos = "http://cran.us.r-project.org")
if(!require(reshape2)) install.packages("reshape2",
                                     repos = "http://cran.us.r-project.org")
if(!require(ggiraph)) install.packages("ggiraph",
                                     repos = "http://cran.us.r-project.org")
if(!require(RColorBrewer)) install.packages("RColorBrewer",
                                     repos="http://cran.us.r-project.org")
if(!require(leaflet)) install.packages("leaflet",
                                     repos = "http://cran.us.r-project.org")
if(!require(plotly)) install.packages("plotly",
                                     repos = "http://cran.us.r-project.org")
if(!require(geojsonio)) install.packages("geojsonio",
                                     repos = "http://cran.us.r-project.org")
if(!require(highcharter)) install.packages("highcharter",
                                     repos = "http://cran.us.r-project.org")
if(!require(tidyr)) install.packages("tidyr",
                                     repos="http://cran.us.r-project.org")
if(!require(knitr)) install.packages("knitr",
                                     repos = "http://cran.us.r-project.org")
if(!require(tinytex)) install.packages("tinytex",
                                     repos = "http://cran.us.r-project.org")
if(!require(sf)) install.packages("sf",
                                     repos = "http://cran.us.r-project.org")
if(!require(kableExtra)) install.packages("kableExtra",
                                     repos = "http://cran.us.r-project.org")

```

## Lecture de notre jeu de données

Utilisation du package `readxl` pour lire le contenu du fichier de notre jeu de données «inflation.xls» et renvoyer la première partie du jeu de données.

```
library(readxl)
library(kableExtra)
#Utilisation du package readxl pour lire le contenu du fichier excel de notre dataset
inflation <- read_excel("input_data/inflation.xls")
#Renvoier la première partie (5 lignes) du jeu de données.
head(inflation)
```

```
## # A tibble: 6 x 44
##   Inflat~1 '1980' '1981' '1982' '1983' '1984' '1985' '1986' '1987' '1988' '1989'
##   <chr>    <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 <NA>    <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
## 2 Afghani~ no da~ no da~ no da~ no da~ no da~ no da~ no da~ no da~ no da~ no da~
## 3 Albania no da~ no da~ no da~ no da~ no da~ no da~ no da~ no da~ no da~ no da~
## 4 Algeria 9.699~ 14.6  6.599~ 7.799~ 6.299~ 10.4  14    5.900~ 5.900~ 9.199~
## 5 Angola  46.70~ 1.399~ 1.8    1.8    1.8    1.8    1.8    1.8    1.8    1.8
## 6 Antigua~ 19    11.5  4.200~ 2.299~ 3.799~ 1      0.5    3.600~ 6.799~ 4.400~
## # ... with 33 more variables: '1990' <chr>, '1991' <chr>, '1992' <chr>,
## #   '1993' <chr>, '1994' <chr>, '1995' <chr>, '1996' <chr>, '1997' <chr>,
## #   '1998' <chr>, '1999' <chr>, '2000' <chr>, '2001' <chr>, '2002' <chr>,
## #   '2003' <chr>, '2004' <chr>, '2005' <chr>, '2006' <chr>, '2007' <chr>,
## #   '2008' <chr>, '2009' <chr>, '2010' <chr>, '2011' <chr>, '2012' <chr>,
## #   '2013' <chr>, '2014' <chr>, '2015' <chr>, '2016' <chr>, '2017' <chr>,
## #   '2018' <chr>, '2019' <chr>, '2020' <chr>, '2021' <chr>, '2022' <chr>, ...
```

## Utilisation d'un fichier geojson (du package "geojsonio")

Pour récupérer les coordonnées spatial de chaque pays pour qu'on puisse les affichées sur la map

```
library(geojsonio)
worldcountry = geojson_read("input_data/50m.geojson", what = "sp")
```

## Pré-traitement de nos données :

Pour une utilisation facile & simple ultérieurement de nos données, nous avons mis les années dans une liste en tant que chaîne de caractère ainsi que de reformater notre jeu de données dans un nouveau dataframe appelé “inf\_data” Enfin nous avons mis les 5 premières lignes de notre nouveau dataframe

```
library(dplyr)
library(tidyr)
library(kableExtra)
#Créer un vecteur contient les années dans notre jeu de données
year<-c(1980:2022)
year<-as.character(year)
#Reformater notre jeu de données d'une façon que nous affichons les taux d'inflation
#par rapport a chaque pays et a chaque années (nous avons rassembler les années
#dans une seule colonne <Year>)
inf_data <- inflation %>% gather(year,key = "Year",value="InflationRate")
names(inf_data)<-c("region","year","inflation")
inf_data$year<-as.integer(inf_data$year)
head(inf_data) %>%
  kable() %>%
  kable_styling(latex_options = "hold_position")
```

region	year	inflation
NA	1980	NA
Afghanistan	1980	no data
Albania	1980	no data
Algeria	1980	9.6999999999999993
Angola	1980	46.700000000000003
Antigua and Barbuda	1980	19

Nous avons mis un sommaire des statistiques de notre nouveau jeu de données

```
inf_data %>%
  summary()
```

```
##      region          year      inflation
## Length:9804      Min.    :1980  Length:9804
## Class :character 1st Qu.:1990  Class :character
## Mode  :character Median  :2001  Mode  :character
##                      Mean    :2001
##                      3rd Qu.:2012
##                      Max.    :2022
```

## Pré-traitement des noms des pays et des régions économiques

Nous avons créé un nouveau variable nommé “country” qui contient une liste des noms de pays “region” à partir de notre nouveau dataset “inf\_data” Ensuite nous avons créé aussi une liste qui contient les régions économiques à travers le monde. (c’est un peu manuel mais c’est efficace)

```
pays <- c(inf_data$region)
unions<-c("Major advanced economies (G7)","European Union",
```

```
"Emerging and Developing Europe","ASEAN-5",
"Commonwealth of Independent States",
"Emerging and Developing Asia","Latin America and the Caribbean",
"Middle East, North Africa, Afghanistan, and Pakistan")
```

## Données spatiales des pays à travers le monde

Nous avons utilisé le fichier “50m.geojson” (téléchargé depuis <src : <https://datahub.io/core/geo-countries>>) pour récupérer les données spatiales de chaque pays afin de les utiliser dans l’affichage de la carte. Voici la première partie (les 5 premières lignes de notre data frame `plot_map`) qui contient les noms des pays contenant dans notre nouveau jeu de données “inf”

```
plot_map <- worldcountry[worldcountry$ADMIN %in% inf_data$region, ]
```

## Comparaison entre quelques taux d’inflations de quelques différentes nations

Nous avons préparé quelques données des 5 pays (Inde, Chine, Japon, USA, Royaume Uni, France, Algérie) pour faire une comparaison entre ces nations en ce qui concerne le taux d’inflation, vous pouvez remarquer le résultat du dataframe de la nation FRANCE ci-dessous

```
India<-filter(inf_data,region=="India")
India$inflation<-as.numeric(India$inflation)
India$year<-as.numeric(India$year)
China<-filter(inf_data,region=="China, People's Republic of")
Japan<-filter(inf_data,region=="Japan")
US<-filter(inf_data,region=="United States")
UK<-filter(inf_data,region=="United Kingdom")
FR<-filter(inf_data,region=="France")
ALG<-filter(inf_data,region=="Algeria")
tail(FR) %>% kable() %>%
  kable_styling(latex_options = "hold_position")
```

region	year	inflation
France	2017	1.3999999999999999
France	2018	1.2
France	2019	1.5
France	2020	1.7
France	2021	1.7
France	2022	1.8

## ui.R

Un fichier de notre projet qui contient la partie frontend ou ce qu'on appelle l'interface graphique

## server.R

Un fichier qui contient la partie backend de notre application avec des fonction reactives et des graphique ainsi qu'une map cartographique pour une visualisation plus dynamique

```
server <- function(input, output) {  
  
  #Création d'une fonction réactive data_input() qui réagit en fonction de  
#l'année sélectionner dans le slider pour l'affichage des taux d'inflation  
  
  data_input <- reactive({  
    inf_data %>%  
      filter(year == input$slider1)  
  })  
  
  #Création d'une 2eme fonction réactive data_input_ordered qui permet de  
#vérifier la correspondance des noms de pays entre la fonction data_input()  
#et les coordonnées spatial de plot_map  
  data_input_ordered <- reactive({  
    data_input()[order(match(data_input()$region, plot_map$ADMIN)),]  
  })  
  ...  
}
```

## Dessiner les graphes de notre Shiny Dashboard App

### Exemple : France

Nous avons utilisé la bibliothèque “Dplyr” pour faire du Pipe ‘%>%’ et de filtrer les pays pour avoir le résultat affiché dans un graph correspondant au pays choisi, ici on a pris, l'exemple de la France mais dans l'application une liste de tout les pays du dataset est dispo pour visualiser chaque taux d'inflation. Pour le graph, nous avons utilisé le package (ggplot2) avec un theme pour une meilleure qualité de visualisation, ce que vous allez le voir juste en dessous du code suivant

```
library(ggplot2)
library(highcharter)
theme_set(theme_bw(base_size = 16))
df<-inf_data %>% filter(region=="France")
df$inflation<-as.numeric(df$inflation)
df$year<-as.numeric(df$year)
#plotting the data
hchart(df, "line",color="#4682B4",hcaes(x=year,y=inflation)) %>%
  hc_exporting(enabled = TRUE) %>%
  hc_tooltip(crosshairs = TRUE, backgroundColor = "#FCF5C5",
    shared = FALSE, borderWidth = 2) %>%
  hc_title(text="Diagramme de série chronologique des taux d'inflation",
    align="center") %>%
  hc_subtitle(text="Source de données: IMF",align="center") %>%
  hc_add_theme(hc_theme_elementary())
```

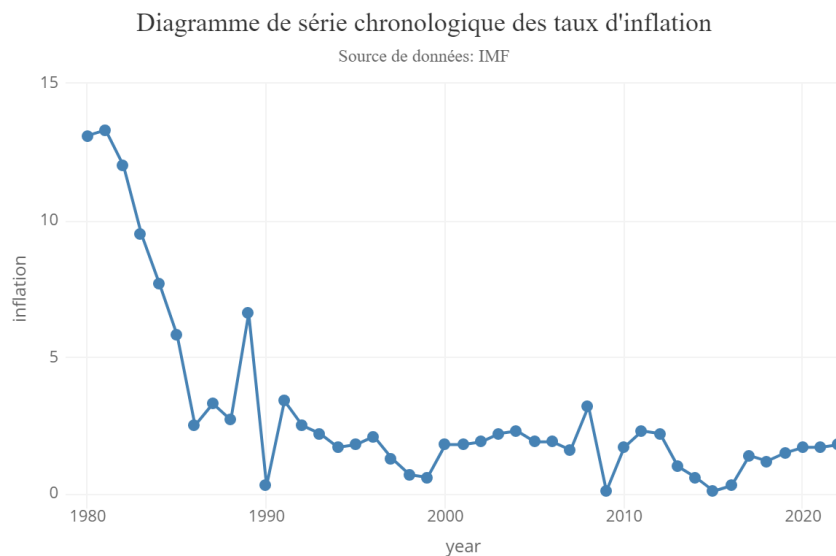


Figure 1: Taux d'inflation en France

## Comparaison de quelques taux d'inflation qui correspond a quelques pays (Choix personnel)

Avec l'utilisation du thème `hc_add_series` du package 'ggplot2' pour une meilleure visualisation des données

```
library(ggplot2)
library(highcharter)
theme_set(theme_bw(base_size = 16))
highchart() %>%
  hc_xAxis(categories=inf_data$year) %>%
  hc_exporting(enabled = TRUE) %>%
  hc_add_series(name = "India", data = India$inflation) %>%
  hc_add_series(name = "USA", data = US$inflation) %>%
  hc_add_series(name = "UK", data = UK$inflation) %>%
  hc_add_series(name = "China", data = China$inflation) %>%
  hc_add_series(name = "France", data = FR$inflation) %>%
  hc_add_series(name="Japan",data=Japan$inflation) %>%
  hc_add_series(name="Algeria",data=ALG$inflation) %>%
  #Pour ajouter les couleurs
  hc_colors(c("red","blue","brown","purple","darkpink","orange","green")) %>%
  hc_add_theme(hc_theme_elementary())
```

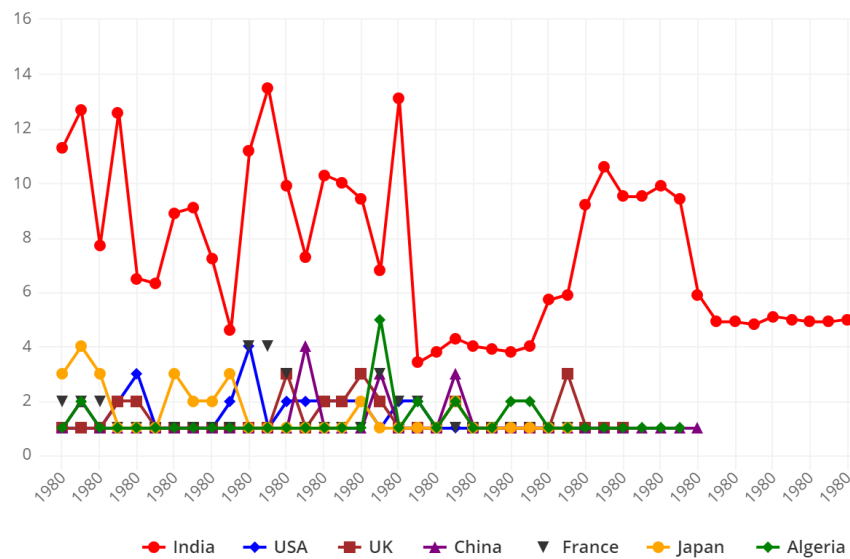


Figure 2: Comparaison des taux d'inflation



## Diagramme chronologique des taux d'inflation dans les plus grandes régions économiques du monde

### Exemple : Union Europeen

Pour afficher l'évolution du taux d'inflation en fonction des régions économiques dans le monde au cours des années (Nous avons pris l'exemple de l'union europeen)

```
library(ggplot2)
library(highcharter)
theme_set(theme_bw(base_size = 16))
union<-inf_data %>% filter(region=="European Union")
union$year<-as.numeric(union$year)
union$inflation<-as.numeric(union$inflation)
#tracage des graphs
hchart(union,hcaes(x=year,y=inflation),
       type="area",color="#2B1F97") %>%
  hc_exporting(enabled = TRUE) %>%
  hc_tooltip(crosshairs = TRUE, backgroundColor = "#FCFFC5",
            shared = TRUE, borderWidth = 2) %>%
  hc_title(text="Diagramme chronologique des taux d'inflation des régions
              économiques dans le monde",align="center") %>%
  hc_subtitle(text="Source de données: IMF",align="center") %>%
  hc_add_theme(hc_theme_elementary())
```

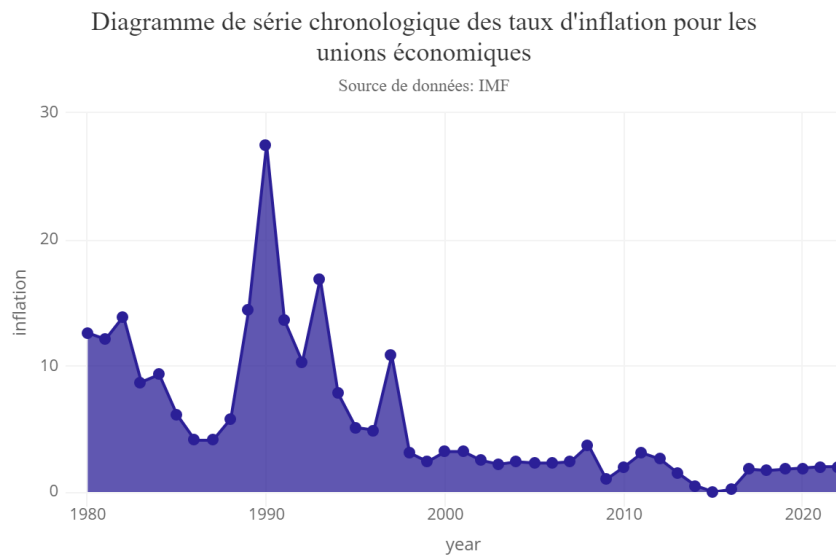


Figure 3: Taux d'inflation de l'union europeen

## Diagramme de série chronologique des taux d'inflation a travers le monde

Un graph qui permet d'afficher l'évolution du taux d'inflation dans le monde entier au fil des années

```
library(ggplot2)
library(highcharter)
theme_set(theme_bw(base_size = 16))
world<-inf_data %>% filter(region=="World")
world$year<-as.numeric(world$year)
world$inflation<-as.numeric(world$inflation)

#traçage des graphs
hchart(world,hcaes(x=year,y=inflation),type="area",color="#4B0082") %>%
  hc_exporting(enabled = TRUE) %>%
  hc_tooltip(crosshairs = TRUE, backgroundColor = "#FCFFC5", shared = TRUE, borderWidth = 2) %>%
  hc_title(text="Diagramme chronologique des taux d'inflation a travers le
             monde",align="center") %>%
  hc_subtitle(text="Source de données: IMF",align="center") %>%
  hc_add_theme(hc_theme_elementary())
```

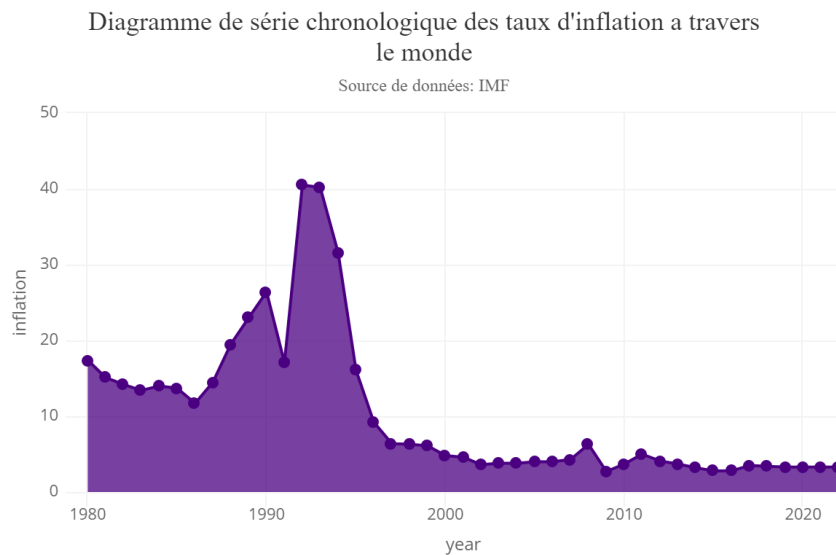


Figure 4: Taux d'inflation du monde

Un graph cartographique qui permet de visualiser a l'aide du slider (pour la sélection des années) le taux d'inflation de chaque pays

```
library(leaflet)

mymap <- leaflet() %>%
  addTiles() %>%
  addPolygons(data = plot_map, color = "blue",
    label = ~paste0("Pays : ", ADMIN," - Taux d'inflation : ",
      as.numeric(inf_data$inflation)),
    weight = 1,
    highlightOptions = highlightOptions(color = "red",
      bringToFront = TRUE,
      weight = 2),
    labelOptions = labelOptions(
      style = list("font-weight" = "normal", padding = "3px 8px"),
      textsize = "15px",
      direction = "auto")
  )
```

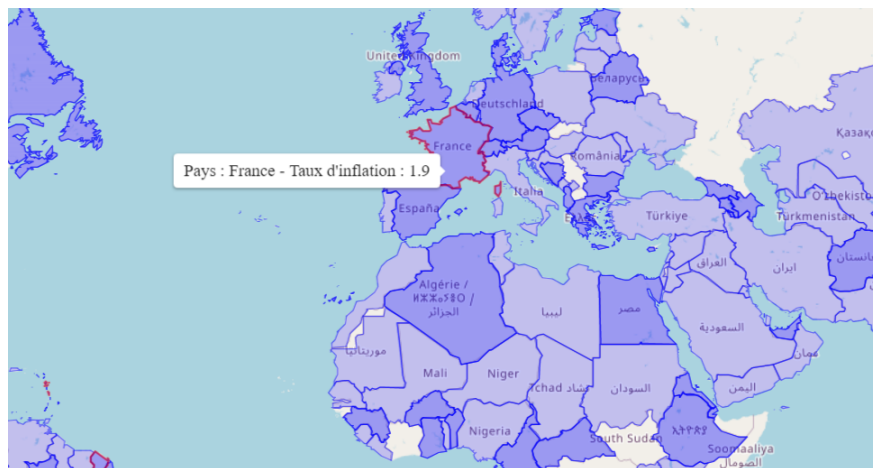


Figure 5: Carte carthographique