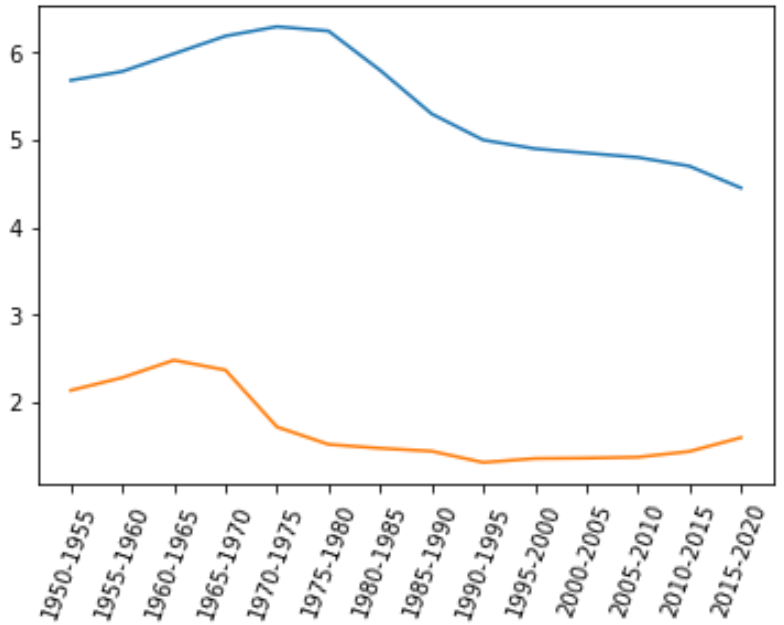
Diplomado en Ciencia de Datos UNAM
Modulo 14 Data Storytelling
Septiembre de 2023

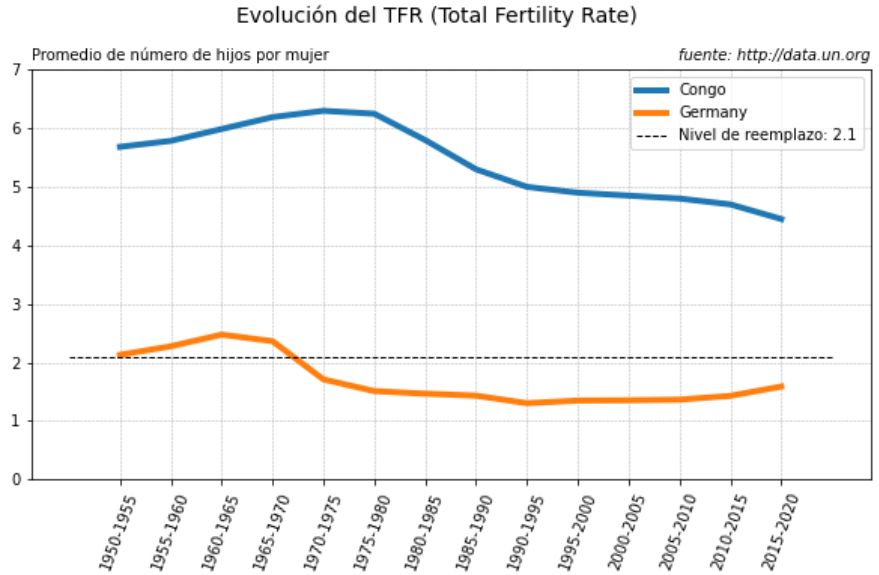
Contenido

1. - Página 1 :
   - La gráfica del ejercicio 1 y su respuesta a la pregunta.
   - La gráfica del ejercicio 2 y su respuesta a la pregunta.
2. - Página 2 :
   - La gráfica final del TFR que Ud. obtuvo y una descripción de la misma.

Describa lo que ve en la gráfica que acaba de obtener

Ejercicio 1

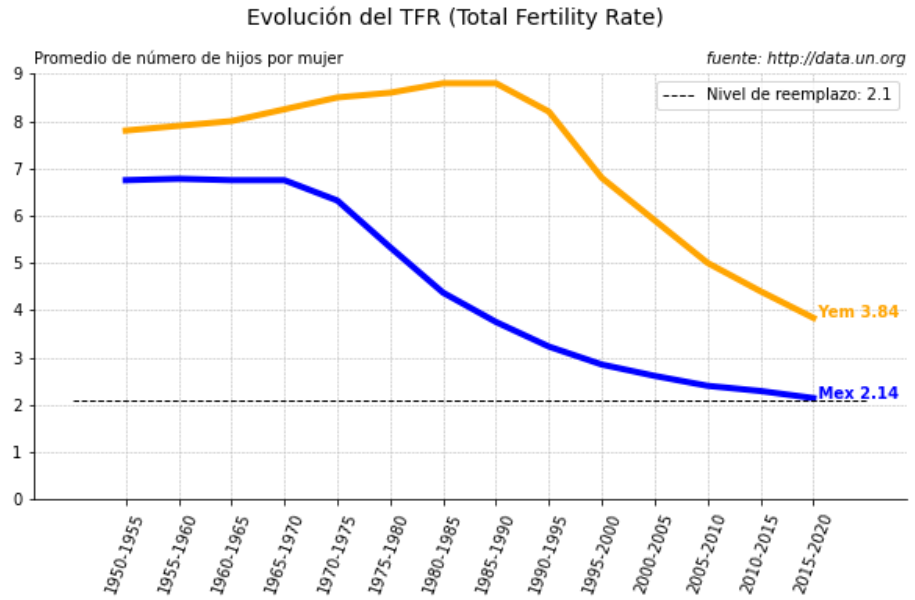


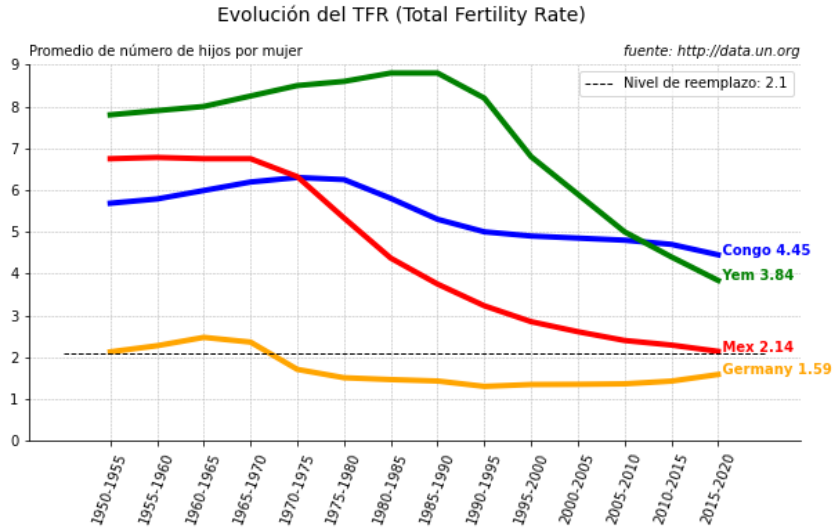


- La TFR de Congo se ha mantenido desde 1955 hasta 2020 unas 3x veces por encima de la de Alemania
- La TFR de ambos países (Congo y Alemania) tuvo un máximo entre 1966 y 1979 y comenzaron un pronunciado descenso en los años 80
- La TFR de Alemania tuvo un mínimo histórico en los años 1990-1995 con TFR=1.3 y comenzó a subir hasta llegar a un TFR=1.59 en el 2020
- A pesar de tener en general una tendencia decreciente, la TFR de Congo solo ha bajado 2.1 en 70 años

Describa lo que ve en la gráfica que acaba de obtener

Ejercicio 2

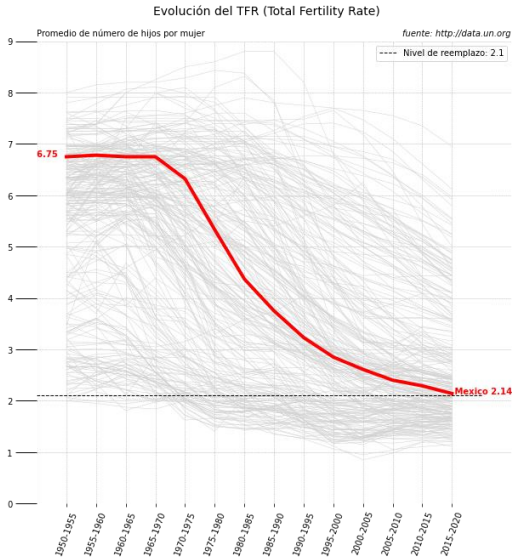


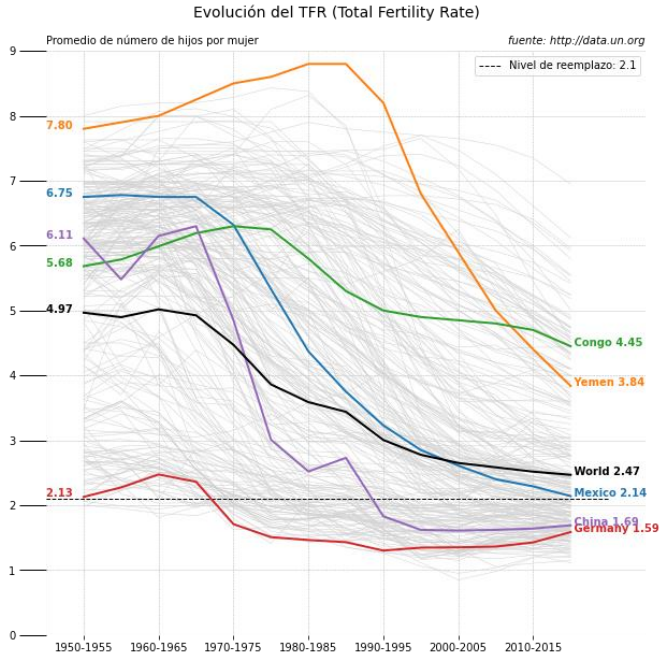


- Los países con menor PIB tienen una mayor TFR que aquellos de menor PIB,
- La TFR de los 3 países (Congo, Alemania, México Yemen) tuvo un máximo entre 1966 y 1979 y comenzaron un pronunciado descenso en los años 80. En cambio, Yemen tuvo su máximo en 1990 y a partir de ahí un descenso tan pronunciado que su TFR ahora es menor que la de Congo

Describa lo que ve en la gráfica que acaba de obtener

Ejercicio 3





- La TFR de México siempre se ha mantenido en un nivel “promedio” con respecto al resto de países
- Yemen es el país que registra el máximo histórico de TFR pero también uno de los descensos más pronunciados desde1 995 hasta colocarse en niveles medios altos de TFR a nivel mundial en el 2020
- China a pasado de tener una TFR de 6.11 normal en países poco desarrollados en 1960 vs un de 1.69 en 2020 lo que concuerda totalmente con el desarrollo económico chino durante los últimos 60 años.
- La TFR mundial ha disminuido en aproximadamente 3.5 hijos por mujer en los últimos 40 años

¿Con este análisis es posible responder a las preguntas que hicimos al principio?

Ejercicio 4

1. ¿Los países ricos están recuperando su TFR?

No se si recuperando es la palabra más adecuada. Algunos países ricos tuvieron sus niveles más basjo de TFR en 1990-2000, pero en general su TFR oscila entre 1.3 y 1.6

2. ¿Cómo es el TFR en países con un bajo PIB per cápita?

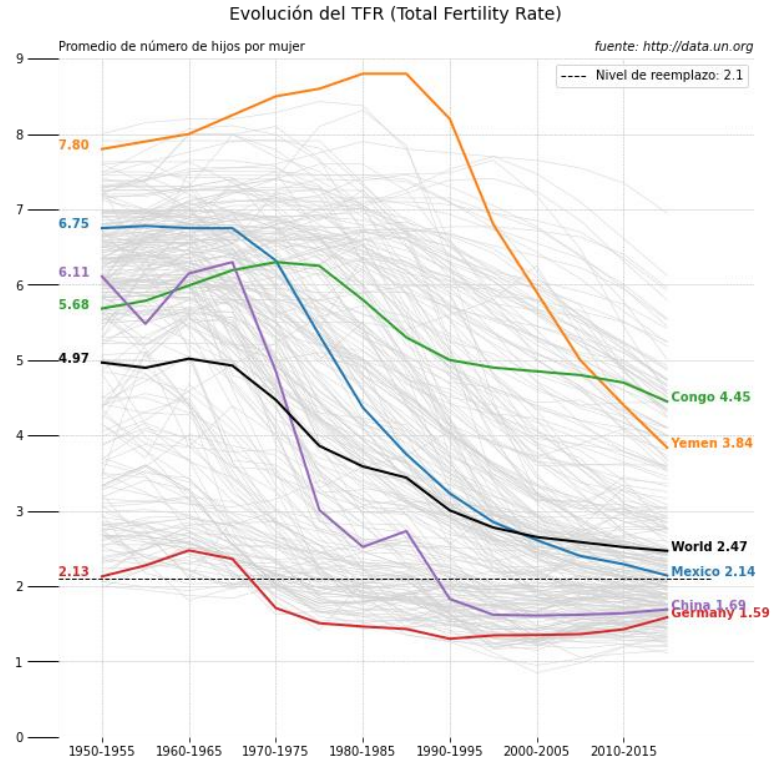
La relación entre PIB y TFR es inversamente proporcional siempre

3. ¿Países en desarrollo como China y Brasil están estabilizando sus poblaciones?

Sin duda, mucho más China que Brasil, quizá por la política del “hijo único que se implantó durante los años 80, 90 y principios del 2000”

4. ¿Cómo ha sido la evolución de la población en México con respecto de otros países?

La TFR de México siempre se ha mantenido en un nivel “promedio” con respecto al resto de países



Mod14_AnalisisPoblacion

September 29, 2023

1 Práctica 1: Análisis de la población mundial

2 Alumno: Ibarra Ramírez Sergio

Módulo 14: Storytelling en ciencia de datos

Diplomado en Ciencia de Datos

DGTIC, Universidad Nacional Autónoma de México

2.1 Introducción

Para lograr una excelente visualización de datos se debe tener: **mucha curiosidad** e interés en **muchos temas variados** que pueden parecer poco relacionados entre ellos: matemáticas, demografía, epidemiología, economía, deportes, ventas por internet, historia, psicología y muchos etcéteras.

Durante el proceso de esta visualización la vida tenderá a convertirse en un **caos intelectual**, pero **sistemático y emocionante**.

La **visualización de datos** es muy importante, pues es posible que sea **lo único que vean tus interlocutores**: cliente, colega, jefe, tutor, jurado, lectores de un periódico, público en una conferencia, … y probablemente a ellos no les interese mucho los datos numéricos o los algoritmos usados en su análisis.

El público a quien deseas transmitir tus hallazgos olvidarán muy pronto los numéros; pero si introduces tus hallazgos mediante un relato que cuente la historia de los datos, es posible que ellos se lleven un buen sabor de boca y recuerden la información recibida por mucho más tiempo e incluso tomen acciones.

2.1.1 El apocalipsis demográfico.

- Algunas historias apocalípticas cuentan que el incremento en el nacimiento de seres humanos en diferentes regiones del mundo, sobre todo en aquellas con poco desarrollo económico, hará que en las próximas dos décadas tengamos que convivir con 9 mil millones de personas en nuestro planeta.

- También dicen que esa es la razón de que la tierra tenga que soportar hoy en día 7.8 mil millones de personas. Vea por ejemplo: población mundial actual.
- ¡¿El crecimiento seguirá de manera exponencial?!

2.1.2 Tasas de fertilidad y reemplazo

- La tasa de fertilidad (TFR: *The total fertility rate*) es una variable demográfica que muestra el número promedio de hijos que nacerían por mujer si todas las mujeres vivieran hasta el final de sus años fértiles y dieran a luz de acuerdo con la tasa de fecundidad promedio para cada edad. Véase Total Fertility Rate.
- La fecundidad de reemplazo (NR: *Net replacement rate*) se refiere a la fecundidad mínima necesaria para que una población cerrada se mantenga indefinidamente en el tiempo sin disminuir su volumen, suele tener un valor de 2.1 hijos por mujer como promedio. Véase Sub-replacement fertility
  - Si este número es mucho menor que 2.1, entonces la población se reduce con el tiempo.
  - Pero si es mucho mayor que 2.1, se tendrá a mucha gente joven, lo cual podría causar problemas, como violencia y crimen.

- En el libro *The rational optimist: How prosperity evolves* de Math Ridley (2010), se dice:
  - "En promedio, la fertilidad en los países ricos es baja, pero en años anteriores ha tenido un incremento. En países pobres se muestra un decrecimiento en la fertilidad. Estas dos tendencias complementarias indican que la **razón de fertilidad se va a estabilizar en 2.1 en pocas décadas**, de tal manera que la población **no pasará de 9 mil millones**." (soportado con datos de la UN y The world bank)

Gráfica del libro de *The rational optimist: How prosperity evolves*, Math Ridley (2010)

2.2 Objetivo

- Extraer información importante de un conjunto de datos y resaltarla mediante algunas técnicas de visualización.
- Construir una visualización que permita comparar el cambio en el Total Fertility Rate (TFR) en función del tiempo.
- Con esta visualización tratar de responder las siguientes preguntas:
  1. ¿Los países ricos están recuperando su TFR?
  2. ¿Cómo es el FR en países con un bajo PIB per cápita?
  3. ¿Países en desarrollo como China y Brasil están estabilizando sus poblaciones?
  4. ¿Cómo ha sido la evolución de la población en México con respecto de otros países?
- Usar Pandas y Matplotlib.

2.3 Importar las bibliotecas

Incluir las bibliotecas necesarias para la lectura de datos y para la visualización.

[1]:
```
import matplotlib.pyplot as plt
import pandas as pd
```

2.4 Descarga de los datos

1. Obtener la información de UNdata con la búsqueda 'fertility rate'
   2. Seleccionar la opción 'Total fertility rate (live births per woman)':
      $ $
3. Seleccionar los rangos de años desde 1950-1955 hasta 2015-2020
   4. Hacer clic en 'Apply Filters' para obtener lo siguiente
      $ $
5. Finalmente descargar el archivo en formato CSV (comprimido)

2.5 Lectura del archivo.

[2]:
```
# Lectura del archivo
TFR = pd.read_csv('UNdata_Export_20211229_175420412.zip')
```

Observe que se puede leer directamente el archivo comprimido. Si descomprime el archivo ZIP obtendrá el archivo en formato CSV el cual también puede leerse como arriba solo cambiando la extensión (`TFR = pd.read_csv('../Datos/UNdata_Export_20211229_175420412.csv')`).

Veamos que contiene el DataFrame:

[3]:
```
TFR # Despliega la parte inicial y final el DataFrame.
```

[3]:

	Country or Area	Year(s)	Variant	Value
0	Afghanistan	2015-2020	Medium	4.555
1	Afghanistan	2010-2015	Medium	5.447
2	Afghanistan	2005-2010	Medium	6.478
3	Afghanistan	2000-2005	Medium	7.182
4	Afghanistan	1995-2000	Medium	7.654
…	…	…	…	…
4041	Zimbabwe	1970-1975	Medium	7.400
4042	Zimbabwe	1965-1970	Medium	7.400
4043	Zimbabwe	1960-1965	Medium	7.300
4044	Zimbabwe	1955-1960	Medium	7.000
4045	Zimbabwe	1950-1955	Medium	6.800

[4046 rows x 4 columns]

Observe que el número total de renglones es de **4046** y se tienen **4** columnas de datos.

Podemos desplegar solamente el principio o solamente el final:

[4]:
```
TFR.head() # Despliega los primeros renglones del DataFrame
```

[4]:
```
  Country or Area    Year(s) Variant  Value
0     Afghanistan  2015-2020  Medium  4.555
1     Afghanistan  2010-2015  Medium  5.447
2     Afghanistan  2005-2010  Medium  6.478
3     Afghanistan  2000-2005  Medium  7.182
4     Afghanistan  1995-2000  Medium  7.654
```

[5]:
```
TFR.tail() # Despliega los últimos renglones del DataFrame
```

[5]:
```
     Country or Area    Year(s) Variant  Value
4041        Zimbabwe  1970-1975  Medium    7.4
4042        Zimbabwe  1965-1970  Medium    7.4
4043        Zimbabwe  1960-1965  Medium    7.3
4044        Zimbabwe  1955-1960  Medium    7.0
4045        Zimbabwe  1950-1955  Medium    6.8
```

2.6 Agrupar por países

Vamos a organizar el DataFrame por países para que su análisis sea más sencillo.

[6]:
```
# Primero se agrupa por país
paises = TFR.groupby('Country or Area')
paises
```

[6]:
```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x00000161EB301600>
```

[9]:
```
print(type(paises)) # paises es un objeto de tipo DataFrameGroupBy
```

```
<class 'pandas.core.groupby.generic.DataFrameGroupBy'>
```

2.6.1 Exploración de la agrupación por países

Vamos a ver que tiene el grupo `países` transformándolo a un DataFrame:

[10]:
```
dummy = pd.DataFrame(paises) # Casting
dummy.index
```

[10]:
```
RangeIndex(start=0, stop=286, step=1)
```

Obsérvese que la agrupación `dummy` solo contiene 286 renglones, lo que indica que tenemos 286 regiones ('Country or Area')

[11]:
```
dummy
```

[11]:
```
                          0  \
0                Afghanistan
1                     Africa
2                    Albania
```

```
3                     Algeria
4              American Samoa
..                       …
281                     World
282  World Bank income groups
283                     Yemen
284                    Zambia
285                  Zimbabwe

                                                      1
0       Country or Area    Year(s) Variant  Value
0…
1       Country or Area    Year(s) Variant  Value
1…
2       Country or Area    Year(s) Variant  Value
2…
3       Country or Area    Year(s) Variant  Value
4…
4       Country or Area    Year(s) Variant  Value
5…
..                                                 …
281       Country or Area    Year(s) Variant  Value…
282                 Country or Area    Year(s) Vari…
283       Country or Area    Year(s) Variant  Value…
284       Country or Area    Year(s) Variant  Value…
285       Country or Area    Year(s) Variant  Value…

[286 rows x 2 columns]
```

El DataFrame `dummy` contiene dos columnas: la 0 y la 1. La 0 nos da el nombre del país o región geográfica y la 1 la información del país.

Vamos a explorar este DataFrame:

[12]:
```
dummy[0] # Imprime la columa correspondiente. Cambie el 0 por 1 y vea lo que␣
↪sucede
```

[12]:
```
0                    Afghanistan
1                         Africa
2                        Albania
3                        Algeria
4                 American Samoa
                  …
281                         World
282    World Bank income groups
283                         Yemen
284                        Zambia
285                      Zimbabwe
```

```
Name: 0, Length: 286, dtype: object
```

[13]:
```
print(type(dummy[0]), type(dummy[1])) # Veamos el tipo de las columas
```

```
<class 'pandas.core.series.Series'> <class 'pandas.core.series.Series'>
```

[14]:
```
dummy.iloc[0] # Con la función iloc accedemos a un renglón particular usando un␣
↪entero
```

[14]:
```
0                                          Afghanistan
1       Country or Area    Year(s) Variant  Value
0…
Name: 0, dtype: object
```

[17]:
```
dummy.iloc[0][1] # Columna correspondiente del renglón, cambie el segundo 0 por␣
↪1 y vea el resultado
```

[17]:

	Country or Area	Year(s)	Variant	Value
0	Afghanistan	2015-2020	Medium	4.555
1	Afghanistan	2010-2015	Medium	5.447
2	Afghanistan	2005-2010	Medium	6.478
3	Afghanistan	2000-2005	Medium	7.182
4	Afghanistan	1995-2000	Medium	7.654
5	Afghanistan	1990-1995	Medium	7.482
6	Afghanistan	1985-1990	Medium	7.469
7	Afghanistan	1980-1985	Medium	7.450
8	Afghanistan	1975-1980	Medium	7.450
9	Afghanistan	1970-1975	Medium	7.450
10	Afghanistan	1965-1970	Medium	7.450
11	Afghanistan	1960-1965	Medium	7.450
12	Afghanistan	1955-1960	Medium	7.450
13	Afghanistan	1950-1955	Medium	7.450

[ ]:
```
print(type(dummy.iloc[0][0]), type(dummy.iloc[0][1])) # tipos de las columnas
```

¿Cómo podemos acceder a la información de un país?

2.6.2 Ejercicio 0.

A continuación el instructor usará España y Suecia como países de trabajo en esta práctica. Usted deberá elegir otros dos países diferentes. **NOTA**: No elija España, Suecia, Yemen ni China.

Para listar los países disponibles haga lo siguiente:

[18]:
```
paises.groups.keys()
```

[18]:
```
dict_keys(['Afghanistan', 'Africa', 'Albania', 'Algeria', 'American Samoa',
'Andorra', 'Angola', 'Anguilla', 'Antigua and Barbuda', 'Argentina', 'Armenia',
'Aruba', 'Asia', 'Australia', 'Australia/New Zealand', 'Austria', 'Azerbaijan',
```

'Bahamas', 'Bahrain', 'Bangladesh', 'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin', 'Bermuda', 'Bhutan', 'Bolivia (Plurinational State of)', 'Bonaire, Sint Eustatius and Saba', 'Bosnia and Herzegovina', 'Botswana', 'Brazil', 'British Virgin Islands', 'Brunei Darussalam', 'Bulgaria', 'Burkina Faso', 'Burundi', 'Cabo Verde', 'Cambodia', 'Cameroon', 'Canada', 'Caribbean', 'Cayman Islands', 'Central African Republic', 'Central America', 'Central Asia', 'Central and Southern Asia', 'Chad', 'Channel Islands', 'Chile', 'China', 'China, Hong Kong SAR', 'China, Macao SAR', 'Colombia', 'Comoros', 'Congo', 'Cook Islands', 'Costa Rica', 'Croatia', 'Cuba', 'Curaçao', 'Cyprus', 'Czechia', "Côte d'Ivoire", "Dem. People's Republic of Korea", 'Democratic Republic of the Congo', 'Denmark', 'Djibouti', 'Dominica', 'Dominican Republic', 'Eastern Africa', 'Eastern Asia', 'Eastern Europe', 'Eastern and South-Eastern Asia', 'Ecuador', 'Egypt', 'El Salvador', 'Equatorial Guinea', 'Eritrea', 'Estonia', 'Eswatini', 'Ethiopia', 'Europe', 'Europe and Northern America', 'Falkland Islands (Malvinas)', 'Faroe Islands', 'Fiji', 'Finland', 'France', 'French Guiana', 'French Polynesia', 'Gabon', 'Gambia', 'Geographic regions', 'Georgia', 'Germany', 'Ghana', 'Gibraltar', 'Greece', 'Greenland', 'Grenada', 'Guadeloupe', 'Guam', 'Guatemala', 'Guinea', 'Guinea-Bissau', 'Guyana', 'Haiti', 'High-income countries', 'Holy See', 'Honduras', 'Hungary', 'Iceland', 'India', 'Indonesia', 'Iran (Islamic Republic of)', 'Iraq', 'Ireland', 'Isle of Man', 'Israel', 'Italy', 'Jamaica', 'Japan', 'Jordan', 'Kazakhstan', 'Kenya', 'Kiribati', 'Kuwait', 'Kyrgyzstan', 'Land-locked Developing Countries (LLDC)', "Lao People's Democratic Republic", 'Latin America and the Caribbean', 'Latvia', 'Least developed countries', 'Lebanon', 'Lesotho', 'Less developed regions', 'Less developed regions, excluding China', 'Less developed regions, excluding least developed countries', 'Liberia', 'Libya', 'Liechtenstein', 'Lithuania', 'Low-income countries', 'Lower-middle-income countries', 'Luxembourg', 'Madagascar', 'Malawi', 'Malaysia', 'Maldives', 'Mali', 'Malta', 'Marshall Islands', 'Martinique', 'Mauritania', 'Mauritius', 'Mayotte', 'Melanesia', 'Mexico', 'Micronesia', 'Micronesia (Fed. States of)', 'Middle Africa', 'Middle-income countries', 'Monaco', 'Mongolia', 'Montenegro', 'Montserrat', 'More developed regions', 'Morocco', 'Mozambique', 'Myanmar', 'Namibia', 'Nauru', 'Nepal', 'Netherlands', 'New Caledonia', 'New Zealand', 'Nicaragua', 'Niger', 'Nigeria', 'Niue', 'No income group available', 'North Macedonia', 'Northern Africa', 'Northern Africa and Western Asia', 'Northern America', 'Northern Europe', 'Northern Mariana Islands', 'Norway', 'Oceania', 'Oceania (excluding Australia and New Zealand)', 'Oman', 'Other non-specified areas', 'Pakistan', 'Palau', 'Panama', 'Papua New Guinea', 'Paraguay', 'Peru', 'Philippines', 'Poland', 'Polynesia', 'Portugal', 'Puerto Rico', 'Qatar', 'Republic of Korea', 'Republic of Moldova', 'Romania', 'Russian Federation', 'Rwanda', 'Réunion', 'Saint Helena', 'Saint Kitts and Nevis', 'Saint Lucia', 'Saint Pierre and Miquelon', 'Saint Vincent and the Grenadines', 'Saint-Barthélemy', 'Saint-Martin (French part)', 'Samoa', 'San Marino', 'Sao Tome and Principe', 'Saudi Arabia', 'Senegal', 'Serbia', 'Seychelles', 'Sierra Leone', 'Singapore', 'Sint Maarten (Dutch part)', 'Slovakia', 'Slovenia', 'Small Island Developing States (SIDS)', 'Solomon Islands', 'Somalia', 'South Africa', 'South America', 'South Sudan', 'South-Eastern Asia', 'Southern Africa', 'Southern Asia', 'Southern Europe',

```
'Spain', 'Sri Lanka', 'State of Palestine', 'Sub-Saharan Africa', 'Sudan',
'Suriname', 'Sustainable Development Goal (SDG) regions', 'Sweden',
'Switzerland', 'Syrian Arab Republic', 'Tajikistan', 'Thailand', 'Timor-Leste',
'Togo', 'Tokelau', 'Tonga', 'Trinidad and Tobago', 'Tunisia', 'Turkey',
'Turkmenistan', 'Turks and Caicos Islands', 'Tuvalu', 'UN development groups',
'Uganda', 'Ukraine', 'United Arab Emirates', 'United Kingdom', 'United Republic
of Tanzania', 'United States Virgin Islands', 'United States of America',
'Upper-middle-income countries', 'Uruguay', 'Uzbekistan', 'Vanuatu', 'Venezuela
(Bolivarian Republic of)', 'Viet Nam', 'Wallis and Futuna Islands', 'Western
Africa', 'Western Asia', 'Western Europe', 'Western Sahara', 'World', 'World
Bank income groups', 'Yemen', 'Zambia', 'Zimbabwe'])
```

2.6.3 Usando `get_group()`

Otra manera de extraer los datos de cada país es usando la función `get_group()`.

Por ejemplo, vamos a obtener la información de dos países con un *PIB per cápita* por arriba de México, por ejemplo España y Suecia. Veáse Países por PIB (nominal) per cápita

[19]:
```
# Se obtienen los datos de Congo y Germany
congo = paises.get_group('Congo')
germany = paises.get_group('Germany')
```

[20]:
```
print(type(congo), type(germany))
```

```
<class 'pandas.core.frame.DataFrame'> <class 'pandas.core.frame.DataFrame'>
```

[21]:
```
congo
```

[21]:

	Country or Area	Year(s)	Variant	Value
784	Congo	2015-2020	Medium	4.450
785	Congo	2010-2015	Medium	4.700
786	Congo	2005-2010	Medium	4.800
787	Congo	2000-2005	Medium	4.850
788	Congo	1995-2000	Medium	4.900
789	Congo	1990-1995	Medium	5.000
790	Congo	1985-1990	Medium	5.300
791	Congo	1980-1985	Medium	5.800
792	Congo	1975-1980	Medium	6.250
793	Congo	1970-1975	Medium	6.300
794	Congo	1965-1970	Medium	6.192
795	Congo	1960-1965	Medium	5.989
796	Congo	1955-1960	Medium	5.786
797	Congo	1950-1955	Medium	5.684

[22]:
```
germany
```

[22]:

	Country or Area	Year(s)	Variant	Value
1358	Germany	2015-2020	Medium	1.586

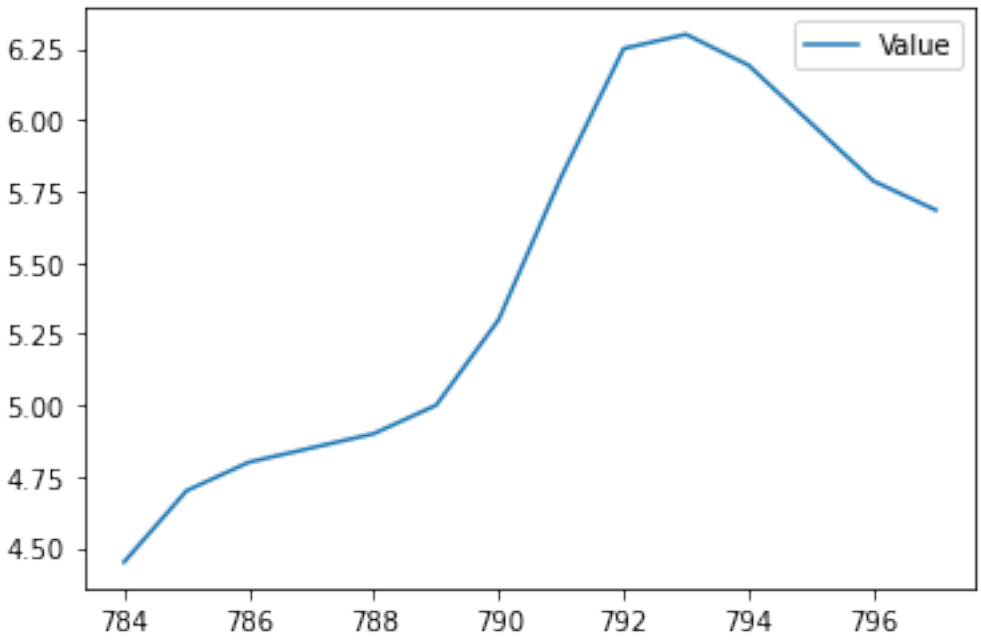
```
1359          Germany  2010-2015  Medium  1.427
1360          Germany  2005-2010  Medium  1.362
1361          Germany  2000-2005  Medium  1.351
1362          Germany  1995-2000  Medium  1.346
1363          Germany  1990-1995  Medium  1.301
1364          Germany  1985-1990  Medium  1.430
1365          Germany  1980-1985  Medium  1.464
1366          Germany  1975-1980  Medium  1.508
1367          Germany  1970-1975  Medium  1.708
1368          Germany  1965-1970  Medium  2.361
1369          Germany  1960-1965  Medium  2.474
1370          Germany  1955-1960  Medium  2.274
1371          Germany  1950-1955  Medium  2.128
```

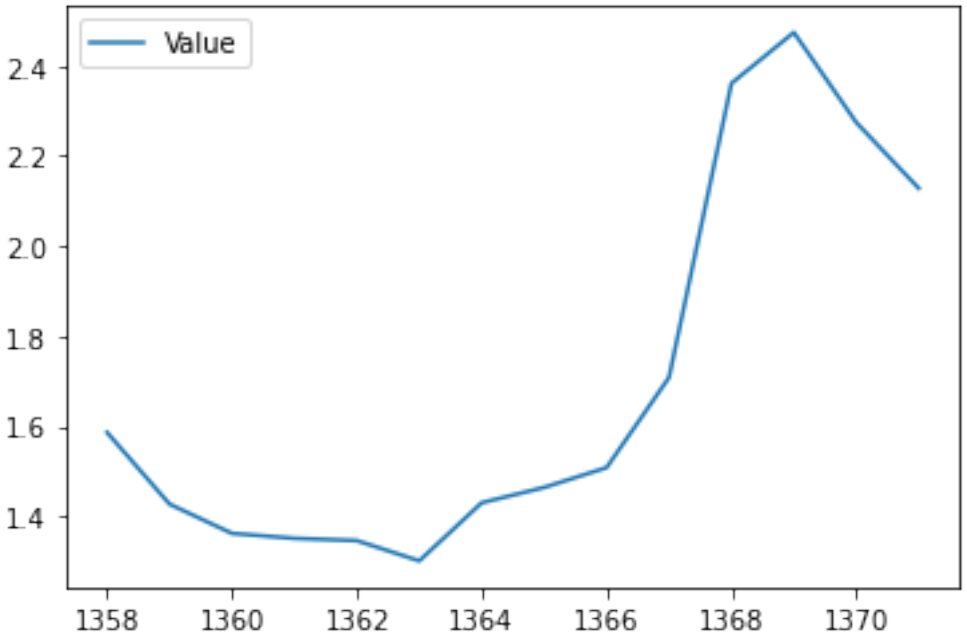
2.7 Graficación usando .plot de DataFrame

Pandas provee de la función plot para hacer un gráfico rápido del DataFrame. Veamos como funciona:

[23]:
```
congo.plot()
germany.plot()
```

[23]:
```
<AxesSubplot:>
```





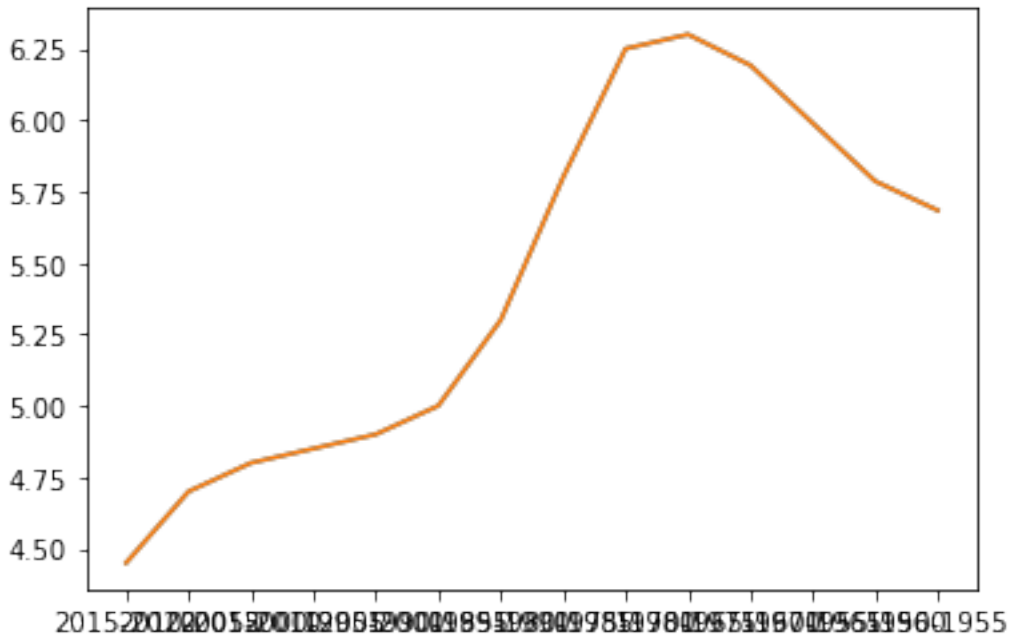
Muchos de los tipos de datos que ofrecen bibliotecas con Pandas, Xarray y otras contienen una función plot que realiza gráficas de manera automática. Casi todas están basadas en Matplotlib. Cada una de esas funciones tiene sus propias funcionalidades y parámentros que pueden modificarse para mejorar las gráficas resultantes.

¿Podría agregar más información a la gráfica anterior?

2.8 Visualización con Matplotlib.

Usando Matplotlib podemos realizar las gráficas España y Suecia como sigue:

[24]:
```
# Se usa la función plot() de Matplotlib
plt.plot(congo['Year(s)'], congo['Value'])
plt.plot(congo['Year(s)'], congo['Value'])

plt.show()
```

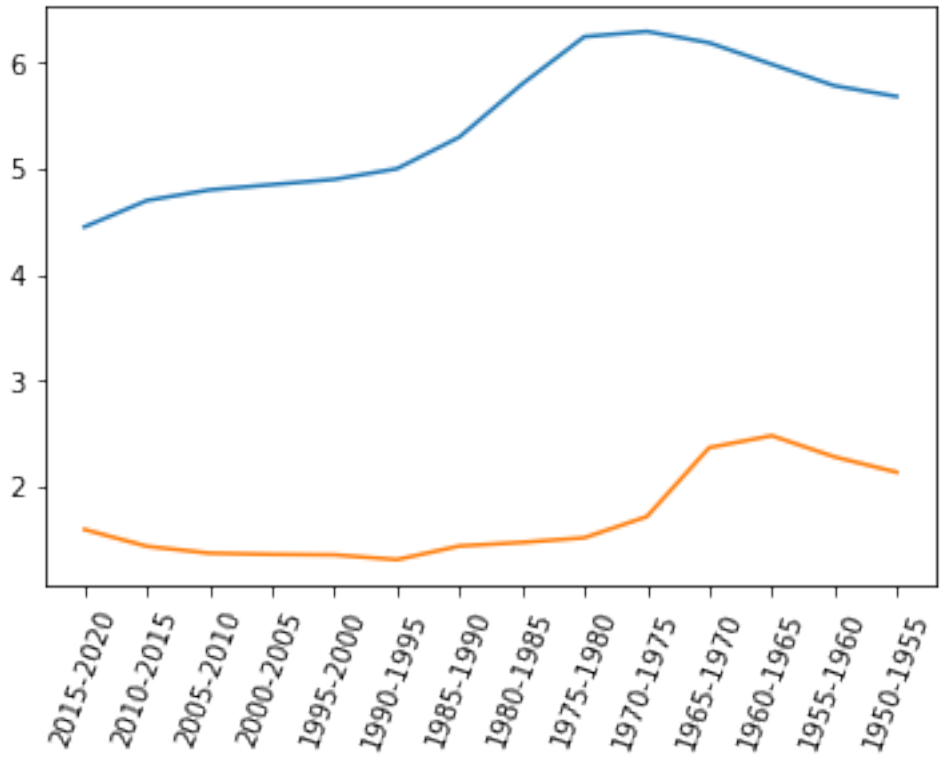


Como se puede observar ambas gráficas aparecen en una sola figura. Pero tenemos muchos problemas para entenderla, empezando con las etiquetas en el eje x (`xtics` y `xlabels`) que aparecen amontonadas.

2.8.1 Rotación de las etiquetas en el eje x.

Rotar los `xtics` para apreciarlos mejor.

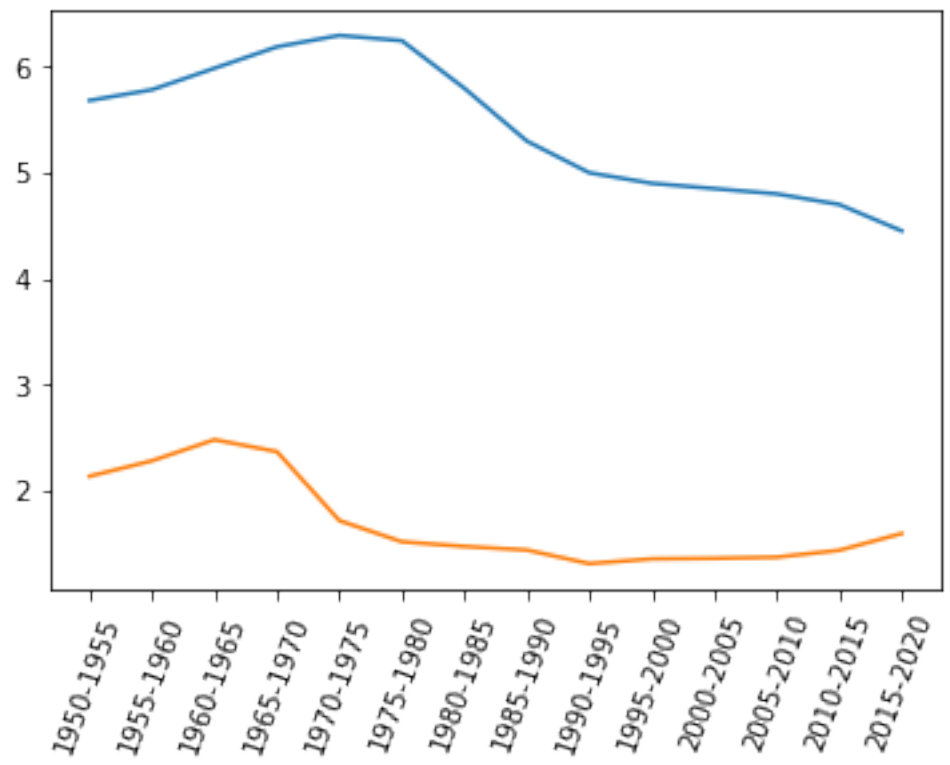
[25]:
```
plt.plot(congo['Year(s)'], congo['Value'])
plt.plot(germany['Year(s)'], germany['Value'])
plt.xticks(rotation=70) # Rotación de 70 grados con respecto al eje x, en␣
 ↪sentido horario.

plt.show()
```



Se observa en el eje x la información de mayor (2015-2020) a menor (1950-1955).

2.8.2 Invertir los datos del eje x

[26]:
```
plt.plot(congo['Year(s)'], congo['Value'])
plt.plot(germany['Year(s)'], germany['Value'])
plt.xticks(rotation=70)
plt.gca().invert_xaxis() # La función gca() obtiene el objeto que controla los
                         # ejes. Una vez teniendo el objeto de los ejes es
                         # posible usar la función invert_xaxis().
plt.show()
```



Ahora la información es un poco más clara y observamos que el comportamiento del TFR desde los años 50's hasta nuestra epoca.

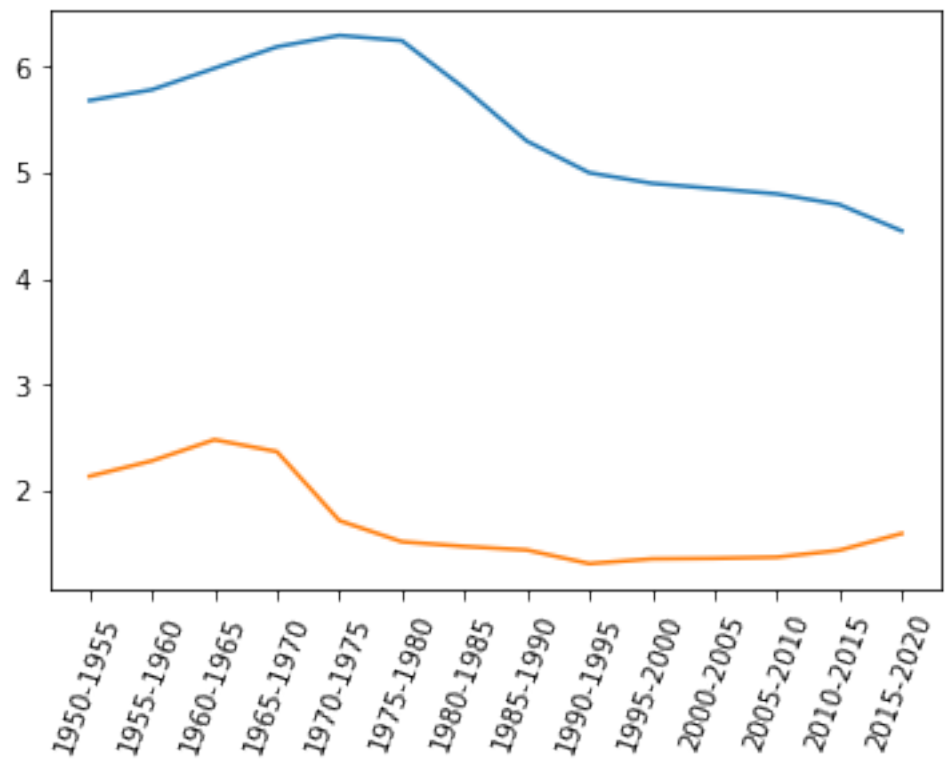
2.8.3 Ejercicio 1.

Describa lo que ve en la gráfica que acaba de obtener.

2.8.4 Resaltar el valor de reemplazo.

Graficamos una línea recta en el valor 2.1, que es considerado el valor de reemplazo (NR).

[27]:
```
plt.plot(congo['Year(s)'], congo['Value'])
plt.plot(germany['Year(s)'], germany['Value'])
#YOUR CODE HERE    # Línea punteada horizontal en y=2.1
plt.xticks(rotation=70)
plt.gca().invert_xaxis()

plt.show()
```

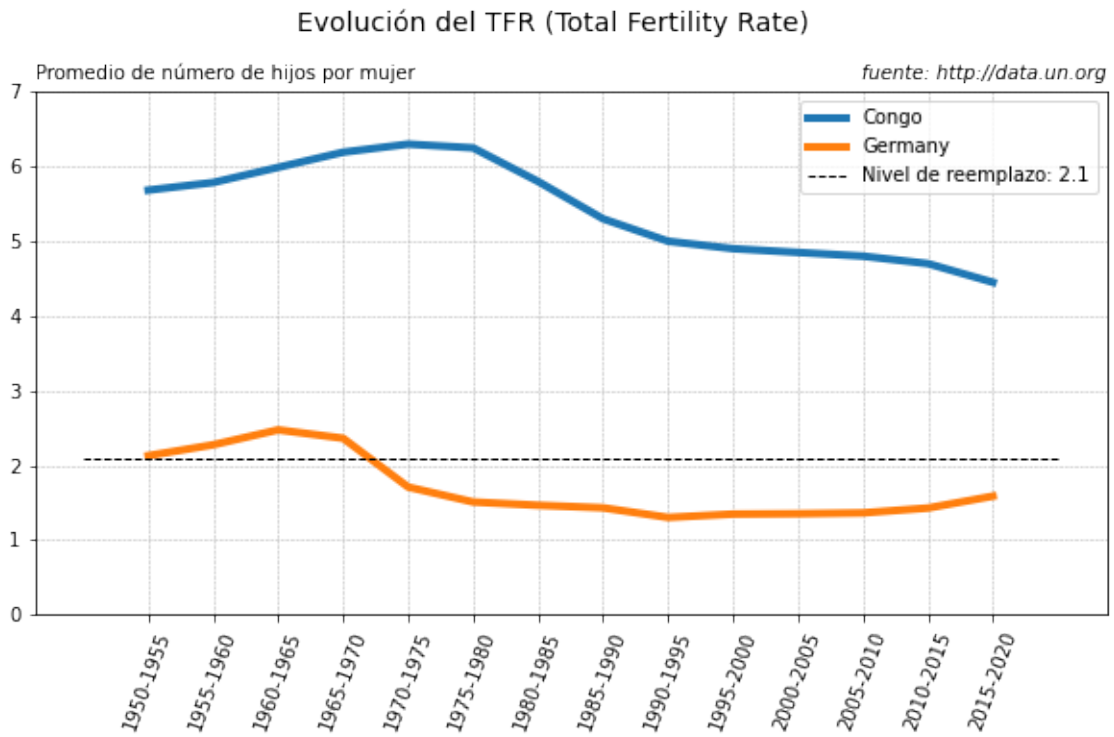


2.8.5 Decoración de la gráfica.

Decoramos la gráfica para tener una primera versión de esta visualización.

[54]:
```
fig = plt.figure(figsize=(10,5)) # Definimos el tamaño de la figura

# Agregamos el ancho de las líneas (lw) y la etiqueta para distinguirlas (label)
plt.plot(congo['Year(s)'], congo['Value'], lw=4.0, label='Congo')
plt.plot(germany['Year(s)'], germany['Value'], lw=4.0, label='Germany')
plt.plot([-1,14],[2.1,2.1], 'k--', lw=1.0, label='Nivel de reemplazo: 2.1')

plt.xticks(rotation=70)
plt.gca().invert_xaxis()
plt.ylim(0,3)              # Límites en el eje y
plt.yticks([0,1,2,3,4,5,6,7])      # Marcas en el eje y
plt.grid(ls='--', lw=0.5) # Rejilla

# Información adicional y títulos
plt.title('Promedio de número de hijos por mujer', loc='left', fontsize=10)
plt.title('fuente: http://data.un.org', loc='right', fontstyle='italic',␣
 ↪fontsize=10)
plt.suptitle('Evolución del TFR (Total Fertility Rate)',y = 1.0, fontsize=14)
```

```
plt.legend() # Generación de la leyenda de la gráfica.

plt.show()
```



2.8.6 Ejercicio 2.

¿Qué opina de esta gráfica con respecto de la del ejercicio 1? ¿Puede detectar algunas *preattentive features* usadas en esta última visualización?

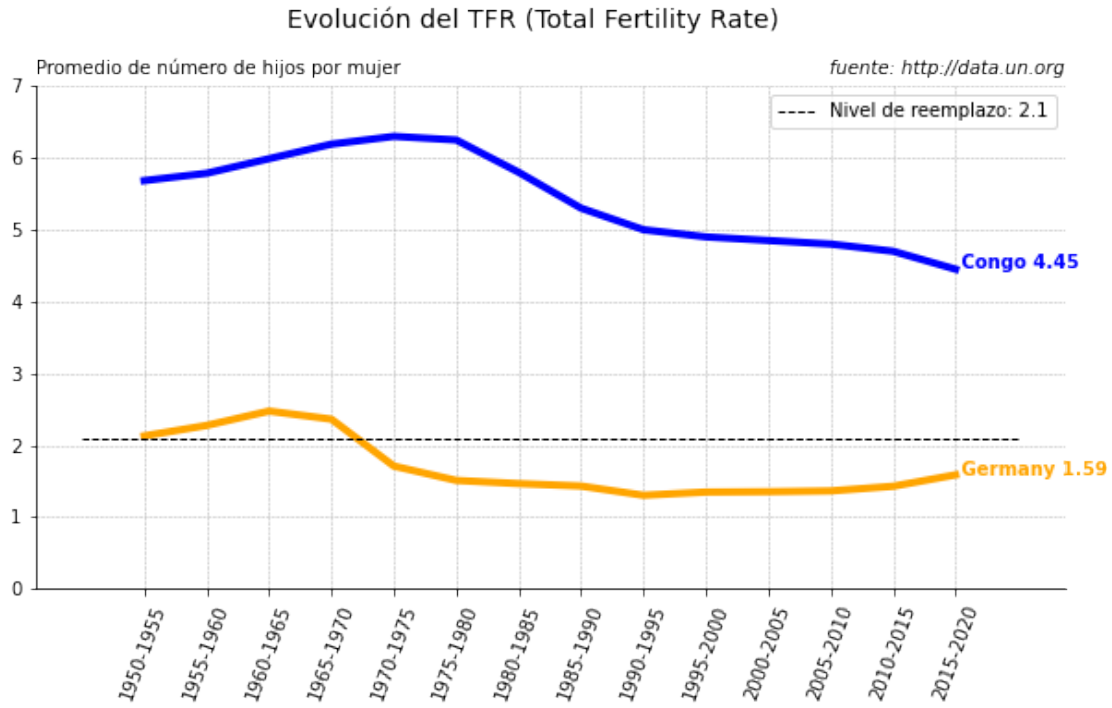
2.8.7 Etiquetado de curvas.

Agregar un texto en el extremo derecho de cada curva para identificarla. El objetivo es eliminar las leyendas de las curvas, pues más adelante tendremos muchas graficas que serán difíciles de distinguir.

Adicionalmente, vamos a quitar protagonismo a las líneas del recuadro de la gráfica (spines), quitaremos dos de ellas.

[55]:
```
fig = plt.figure(figsize=(10,5))

# Definimos colores para las curvas y quitamos las leyendas
congo_color = 'blue'
germany_color = 'orange'
```

```
# Eliminados el label de las gráficas de los países y agregamos un color.
plt.plot(congo['Year(s)'], congo['Value'], lw=4.0, color=congo_color)
plt.plot(germany['Year(s)'], germany['Value'], lw=4.0, color=germany_color)
plt.plot([-1,14],[2.1,2.1], 'k--', lw=1.0, label='Nivel de reemplazo: 2.1')

plt.xticks(rotation=70)
plt.gca().invert_xaxis()
plt.ylim(0,3)
plt.yticks([0,1,2,3,4,5,6,7])
plt.grid(ls='--', lw=0.5)

# Información adicional y títulos
plt.title('Promedio de número de hijos por mujer', loc='left', fontsize=10)
plt.title('fuente: http://data.un.org', loc='right', fontstyle='italic',␣
 ↪fontsize=10)
plt.suptitle('Evolución del TFR (Total Fertility Rate)',y = 1.0, fontsize=14)
plt.legend()

# Obtenemos el valor del FR y lo agregamos al final de cada curva
congo_val = congo['Value'].iloc[0]
germany_val = germany['Value'].iloc[0]
plt.text(x = 0, y = congo_val, s = ' Congo {:1.2f}'.format(congo_val), color =␣
 ↪congo_color, weight = 'bold')
plt.text(x = 0, y = germany_val, s = ' Germany {:1.2f}'.format(germany_val),␣
 ↪color = germany_color, weight = 'bold')

# Quitamos la línea derecha y la de arriba del marco de los ejes
ejes = fig.axes
ejes[0].spines['right'].set_visible(False)
ejes[0].spines['top'].set_visible(False)

plt.show()
```



Esta última gráfica muestra como el número promedio de hijos por mujer se incrementó levemente de 1950 a 1965 para Suecia, para después reducirse por debajo del nivel de reemplazo (NR). Desde entonces ha oscilado un par de veces y parece que a partir de 2005 comienza a estabilizarse en un valor por debajo de 2.1. Algo similar sucede con España, donde el incremento fue un poco más notorio de 1950 hasta 1975, para después bajar y estabilizarse en la década pasada.

¿Qué hechos historicos se podrían correlacionar con estos datos?

A continuación deseamos mostrar que **la razón de nacimientos en todos los países se ha estabilizado**, lo cual nos puede llevar a que la población mundial realmente se estabilice en aproximadamente 9 mil millones de seres humanos. Esta podría ser la historia que se queremos contar.

2.9 Agregar otros países al análisis.

Haremos el mismo análisis para México y Yemen, este último tiene un PIB per cápita por abajo de nuestro país.

[34]:
```
# Extraer los datos de México y Yemen
mex = paises.get_group('Mexico')
yem = paises.get_group('Yemen')
```

2.9.1 Valor máximo del TFR.

Obtener el valor máximo de TFR de ambos países para usarlo en los límites del eje y.

[35]:
```
# Importamos la biblioteca ceil para redondear el valor máximo
from math import ceil

y_maximo = max(mex['Value'].max(), yem['Value'].max()) # El máximo entre dos␣
 ↪países.
yticks = [i for i in range(0,ceil(y_maximo)+1)] # Lista de ticks para el eje y.
print(y_maximo, yticks)
```

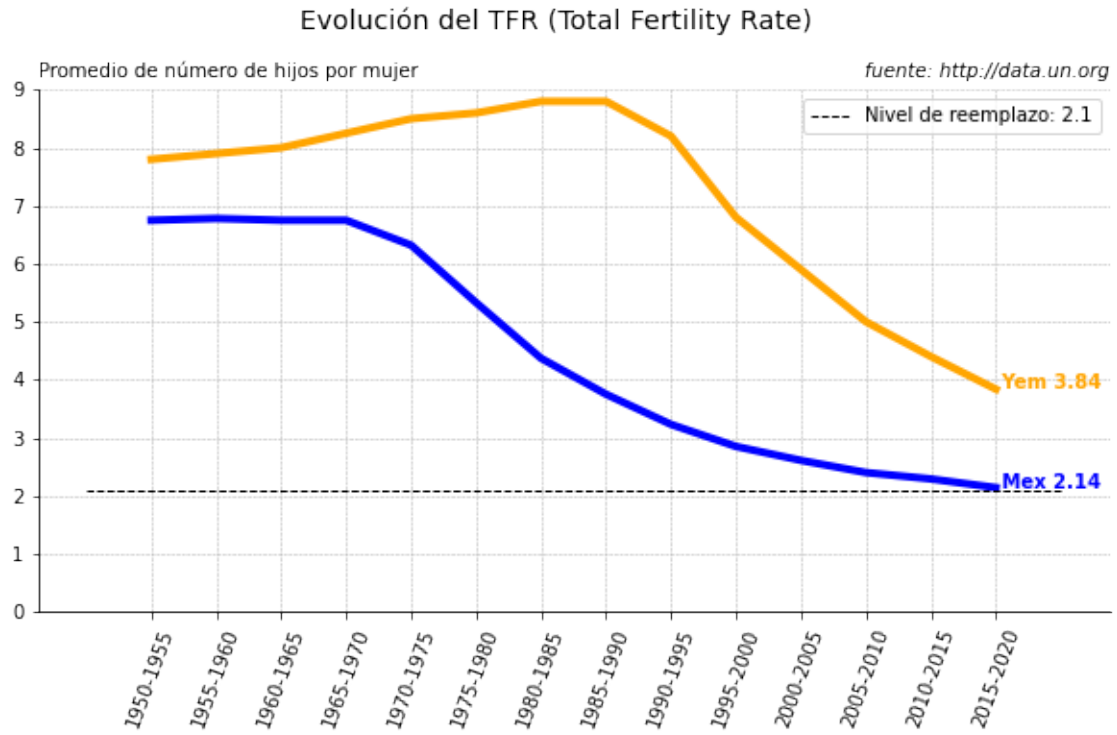
```
8.8 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

2.9.2 Graficación de la nueva información.

Graficar usando el mismo código que usamos para Suecia y España.

[36]:
```
fig = plt.figure(figsize=(10,5))

# Usamos ahora la información de México y Yemen
mex_color = 'blue'
yem_color = 'orange'
plt.plot(mex['Year(s)'], mex['Value'], lw=4.0, c=mex_color)
plt.plot(yem['Year(s)'], yem['Value'], lw=4.0, c=yem_color)
plt.plot([-1,14],[2.1,2.1], 'k--', lw=1.0, label='Nivel de reemplazo: 2.1')

plt.xticks(rotation=70)
plt.gca().invert_xaxis()
plt.ylim(0, y_maximo)    # Usamos el FR máximo
plt.yticks(yticks)       # Usamos los yticks calculados antes
plt.grid(ls='--', lw=0.5)

# Información adicional y títulos
plt.title('Promedio de número de hijos por mujer', loc='left', fontsize=10)
plt.title('fuente: http://data.un.org', loc='right', fontstyle='italic',␣
 ↪fontsize=10)
plt.suptitle('Evolución del TFR (Total Fertility Rate)',y = 1.0, fontsize=14)
plt.legend()

# Agregamos el texto en cada curva
mex_val = mex['Value'].iloc[0]
yem_val = yem['Value'].iloc[0]
plt.text(x = 0, y = mex_val, s = ' Mex {:1.2f}'.format(mex_val), c = mex_color,␣
 ↪weight = 'bold')
plt.text(x = 0, y = yem_val, s = ' Yem {:1.2f}'.format(yem_val), c = yem_color,␣
 ↪weight = 'bold')

# Quitamos la línea derecha y la de arriba del marco de los ejes
ejes = fig.axes
ejes[0].spines['right'].set_visible(False)
ejes[0].spines['top'].set_visible(False)
```

```
plt.show()
```



Observamos en esta gráfica que el TFR en México y Yemen casi siempre han sido mayores al nivel de reemplazo.

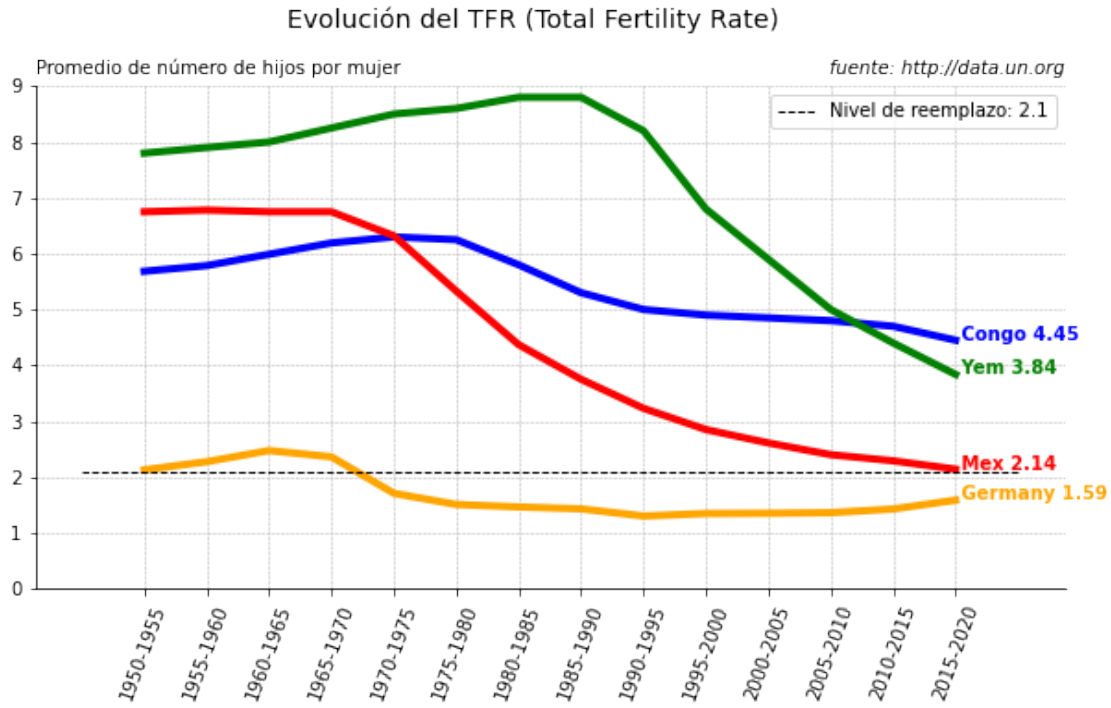
¿Qué más se puede decir de estas curvas con respecto de las de España y Suecia?

2.9.3 Visualización de la información de los cuatro países.

Graficamos los cuatro países juntos para comparar la información.

[40]:
```
fig = plt.figure(figsize=(10,5))

# Definimos todos los colores y quitamos las leyendas
congo_color_color = 'blue'
germany_color = 'orange'
mex_color = 'red'
yem_color = 'green'
plt.plot(congo['Year(s)'], congo['Value'], lw=4.0, c=congo_color)
plt.plot(germany['Year(s)'], germany['Value'], lw=4.0, c=germany_color)
plt.plot(mex['Year(s)'], mex['Value'], lw=4.0, c=mex_color)
plt.plot(yem['Year(s)'], yem['Value'], lw=4.0, c=yem_color)
plt.plot([-1,14],[2.1,2.1], 'k--', lw=1.0, label='Nivel de reemplazo: 2.1')
```

```
plt.xticks(rotation=70)
plt.gca().invert_xaxis()
plt.ylim(0,y_maximo)      # Límites en el eje y (usamos y_maximo)
plt.yticks(yticks)        # Marcas en el eje y
plt.grid(ls='--', lw=0.5) # Rejilla

# Información adicional y títulos
plt.title('Promedio de número de hijos por mujer', loc='left', fontsize=10)
plt.title('fuente: http://data.un.org', loc='right', fontstyle='italic',␣
 ↪fontsize=10)
plt.suptitle('Evolución del TFR (Total Fertility Rate)',y = 1.0, fontsize=14)
plt.legend()

# Agregamos el texto en cada curva
congo_val = congo['Value'].iloc[0]
germany_val = germany['Value'].iloc[0]
mex_val = mex['Value'].iloc[0]
yem_val = yem['Value'].iloc[0]
plt.text(x = 0, y = congo_val, s = ' Congo {:1.2f}'.format(congo_val), c =␣
 ↪congo_color, weight = 'bold')
plt.text(x = 0, y = germany_val, s = ' Germany {:1.2f}'.format(germany_val), c␣
 ↪= germany_color, weight = 'bold')
plt.text(x = 0, y = mex_val, s = ' Mex {:1.2f}'.format(mex_val), c = mex_color,␣
 ↪weight = 'bold')
plt.text(x = 0, y = yem_val, s = ' Yem {:1.2f}'.format(yem_val), c = yem_color,␣
 ↪weight = 'bold')

# Quitamos la línea derecha y la de arriba del marco de los ejes
ejes = fig.axes
ejes[0].spines['right'].set_visible(False)
ejes[0].spines['top'].set_visible(False)

plt.show()
```



Al revisar esta visualización surgen muchas preguntas:

- ¿Cómo se comparan estas gráficas?
- ¿Qué se puede pensar de los países que tienen un TFR que está por arriba del factor NR?
- ¿Cómo podrían relacionarse los hechos históricos y geográficos de esos países con esta información?
- ¿Qué más se observa en estas gráficas?

2.9.4 Visualización de todos los países.

Realizar las gráficas para todos los países de la base de datos.

[41]:
```
# La lista de países se puede obtener usando la función: paises.groups.keys().
paises.groups.keys()
```

[41]:
```
dict_keys(['Afghanistan', 'Africa', 'Albania', 'Algeria', 'American Samoa',
'Andorra', 'Angola', 'Anguilla', 'Antigua and Barbuda', 'Argentina', 'Armenia',
'Aruba', 'Asia', 'Australia', 'Australia/New Zealand', 'Austria', 'Azerbaijan',
'Bahamas', 'Bahrain', 'Bangladesh', 'Barbados', 'Belarus', 'Belgium', 'Belize',
'Benin', 'Bermuda', 'Bhutan', 'Bolivia (Plurinational State of)', 'Bonaire, Sint
Eustatius and Saba', 'Bosnia and Herzegovina', 'Botswana', 'Brazil', 'British
Virgin Islands', 'Brunei Darussalam', 'Bulgaria', 'Burkina Faso', 'Burundi',
'Cabo Verde', 'Cambodia', 'Cameroon', 'Canada', 'Caribbean', 'Cayman Islands',
'Central African Republic', 'Central America', 'Central Asia', 'Central and
Southern Asia', 'Chad', 'Channel Islands', 'Chile', 'China', 'China, Hong Kong
```

SAR', 'China, Macao SAR', 'Colombia', 'Comoros', 'Congo', 'Cook Islands', 'Costa Rica', 'Croatia', 'Cuba', 'Curaçao', 'Cyprus', 'Czechia', "Côte d'Ivoire", "Dem. People's Republic of Korea", 'Democratic Republic of the Congo', 'Denmark', 'Djibouti', 'Dominica', 'Dominican Republic', 'Eastern Africa', 'Eastern Asia', 'Eastern Europe', 'Eastern and South-Eastern Asia', 'Ecuador', 'Egypt', 'El Salvador', 'Equatorial Guinea', 'Eritrea', 'Estonia', 'Eswatini', 'Ethiopia', 'Europe', 'Europe and Northern America', 'Falkland Islands (Malvinas)', 'Faroe Islands', 'Fiji', 'Finland', 'France', 'French Guiana', 'French Polynesia', 'Gabon', 'Gambia', 'Geographic regions', 'Georgia', 'Germany', 'Ghana', 'Gibraltar', 'Greece', 'Greenland', 'Grenada', 'Guadeloupe', 'Guam', 'Guatemala', 'Guinea', 'Guinea-Bissau', 'Guyana', 'Haiti', 'High-income countries', 'Holy See', 'Honduras', 'Hungary', 'Iceland', 'India', 'Indonesia', 'Iran (Islamic Republic of)', 'Iraq', 'Ireland', 'Isle of Man', 'Israel', 'Italy', 'Jamaica', 'Japan', 'Jordan', 'Kazakhstan', 'Kenya', 'Kiribati', 'Kuwait', 'Kyrgyzstan', 'Land-locked Developing Countries (LLDC)', "Lao People's Democratic Republic", 'Latin America and the Caribbean', 'Latvia', 'Least developed countries', 'Lebanon', 'Lesotho', 'Less developed regions', 'Less developed regions, excluding China', 'Less developed regions, excluding least developed countries', 'Liberia', 'Libya', 'Liechtenstein', 'Lithuania', 'Low-income countries', 'Lower-middle-income countries', 'Luxembourg', 'Madagascar', 'Malawi', 'Malaysia', 'Maldives', 'Mali', 'Malta', 'Marshall Islands', 'Martinique', 'Mauritania', 'Mauritius', 'Mayotte', 'Melanesia', 'Mexico', 'Micronesia', 'Micronesia (Fed. States of)', 'Middle Africa', 'Middle-income countries', 'Monaco', 'Mongolia', 'Montenegro', 'Montserrat', 'More developed regions', 'Morocco', 'Mozambique', 'Myanmar', 'Namibia', 'Nauru', 'Nepal', 'Netherlands', 'New Caledonia', 'New Zealand', 'Nicaragua', 'Niger', 'Nigeria', 'Niue', 'No income group available', 'North Macedonia', 'Northern Africa', 'Northern Africa and Western Asia', 'Northern America', 'Northern Europe', 'Northern Mariana Islands', 'Norway', 'Oceania', 'Oceania (excluding Australia and New Zealand)', 'Oman', 'Other non-specified areas', 'Pakistan', 'Palau', 'Panama', 'Papua New Guinea', 'Paraguay', 'Peru', 'Philippines', 'Poland', 'Polynesia', 'Portugal', 'Puerto Rico', 'Qatar', 'Republic of Korea', 'Republic of Moldova', 'Romania', 'Russian Federation', 'Rwanda', 'Réunion', 'Saint Helena', 'Saint Kitts and Nevis', 'Saint Lucia', 'Saint Pierre and Miquelon', 'Saint Vincent and the Grenadines', 'Saint-Barthélemy', 'Saint-Martin (French part)', 'Samoa', 'San Marino', 'Sao Tome and Principe', 'Saudi Arabia', 'Senegal', 'Serbia', 'Seychelles', 'Sierra Leone', 'Singapore', 'Sint Maarten (Dutch part)', 'Slovakia', 'Slovenia', 'Small Island Developing States (SIDS)', 'Solomon Islands', 'Somalia', 'South Africa', 'South America', 'South Sudan', 'South-Eastern Asia', 'Southern Africa', 'Southern Asia', 'Southern Europe', 'Spain', 'Sri Lanka', 'State of Palestine', 'Sub-Saharan Africa', 'Sudan', 'Suriname', 'Sustainable Development Goal (SDG) regions', 'Sweden', 'Switzerland', 'Syrian Arab Republic', 'Tajikistan', 'Thailand', 'Timor-Leste', 'Togo', 'Tokelau', 'Tonga', 'Trinidad and Tobago', 'Tunisia', 'Turkey', 'Turkmenistan', 'Turks and Caicos Islands', 'Tuvalu', 'UN development groups', 'Uganda', 'Ukraine', 'United Arab Emirates', 'United Kingdom', 'United Republic of Tanzania', 'United States Virgin Islands', 'United States of America',

```
'Upper-middle-income countries', 'Uruguay', 'Uzbekistan', 'Vanuatu', 'Venezuela
(Bolivarian Republic of)', 'Viet Nam', 'Wallis and Futuna Islands', 'Western
Africa', 'Western Asia', 'Western Europe', 'Western Sahara', 'World', 'World
Bank income groups', 'Yemen', 'Zambia', 'Zimbabwe'])
```

Como se puede observar, la información es amplia. Si observa con cuidado verá que también hay información definida por regiones geográficas y por otros criterios.

Para ver el total podemos hacer lo siguiente:

[42]:
```
pd.DataFrame(paises).index # Primero transformamos el grupo en DataFrame y luego
                           # obtenemos todos sus índices (renglones)
```

[42]:
```
RangeIndex(start=0, stop=286, step=1)
```

Podríamos entonces graficar **286** curvas. ¿Será esta una buena estrategia?

Para lograr un buen resultado, primero vamos a crear algunas funciones.

Incialización del canvas Primero vamos a definir una función para inicializar el entorno de la gráfica (el canvas).

[43]:
```
def inicializaGrafica(y_maximo, yticks):
    """
    Inicializa algunos parámetros de la figura.

    Parameters
    ----------
    y_maximo : int
        Valor máximo para el eje y.

    yticks : list
        Lista de valores para los ticks en el eje y.
    """
    fig = plt.figure(figsize=(10,10))
    plt.xticks(rotation=70)
    plt.xlim(-2,14,1)
    plt.gca().invert_xaxis()
    plt.ylim(0,y_maximo)
    plt.yticks(yticks)      # Marcas en el eje y
    plt.grid(ls='--', lw=0.5) # Rejilla

    # Información adicional y títulos
    plt.title('Promedio de número de hijos por mujer', loc='left', fontsize=10)
    plt.title('fuente: http://data.un.org', loc='right', fontstyle='italic',␣
↪fontsize=10)
    plt.suptitle('Evolución del TFR (Total Fertility Rate)', y = 0.94,␣
↪fontsize=14)
```

```
    # Se eliminan las líneas del marco de la gráfica
    ejes = fig.axes
    ejes[0].spines['right'].set_visible(False)
    ejes[0].spines['top'].set_visible(False)
    ejes[0].spines['left'].set_visible(False)
    ejes[0].spines['bottom'].set_visible(False)

    # Modificamos algunos parámetros de los ticks en el eje y
    ejes[0].tick_params(axis='y', width=1, length=25)
```

Función para graficar la información de todos los países. Ahora definimos una función para graficar la información de cada país.

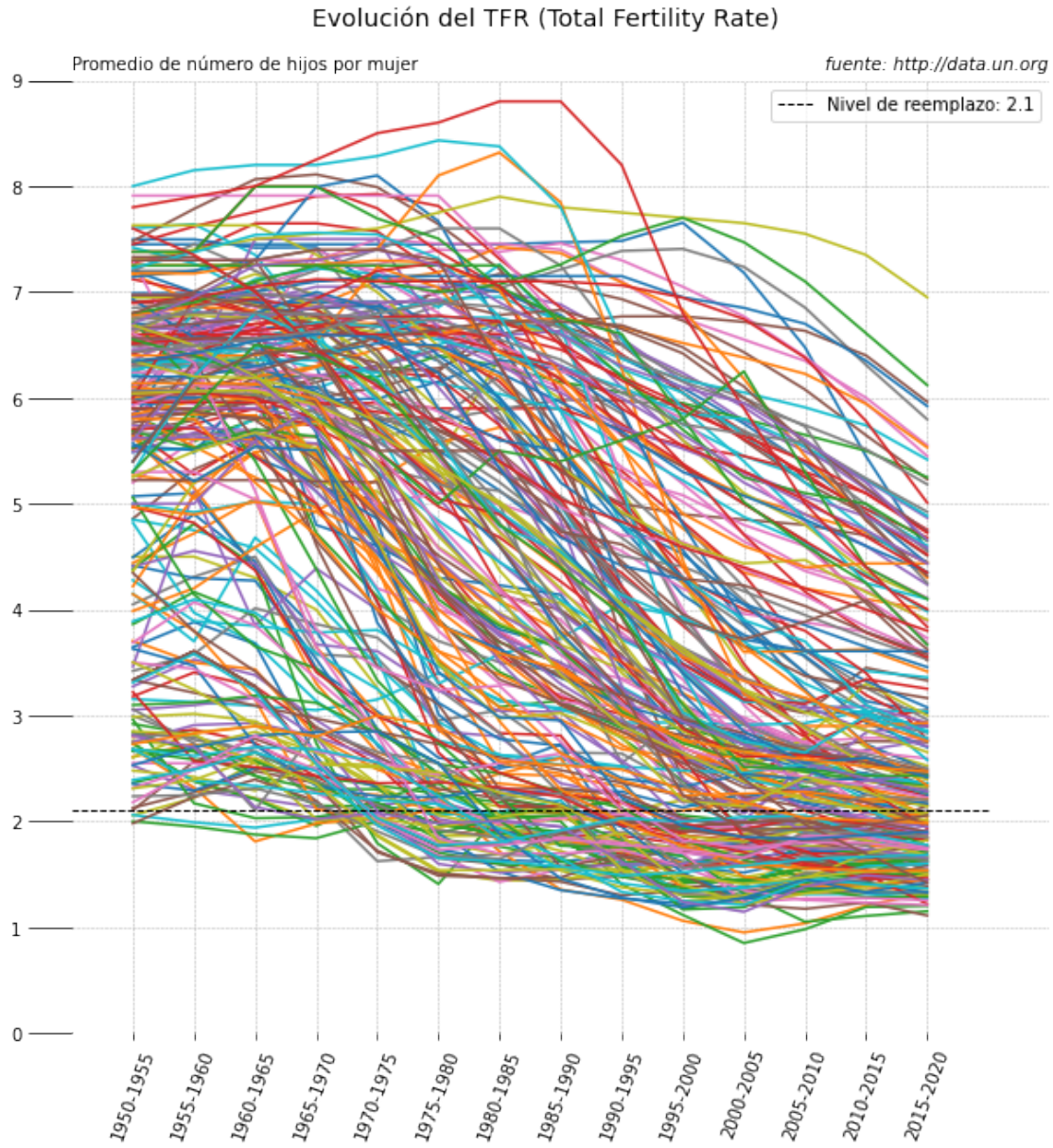
[44]:
```
def graficaTFR(paises, parametros={}):
    """
    Realiza la gráfica de todos los países.

    Parameters
    ----------
    paises : DataFrameGroupBy
        Dataframe generado por GroupBy con la información de los países.

    parametros : dict
        Parámetros para generar la gráfica.
    """
    for p in paises.groups.keys():
        pais = paises.get_group(p)  # Se obtiene el DataFrame del país
        plt.plot(pais['Year(s)'], pais['Value'], **parametros)

    # Al final de todas las gráficas ponemos la del nivel de reemplazo
    # para que apareza sobre todas ellas y se note mejor (también es posible␣
 ↪usar zorder)
    plt.plot([-1,14],[2.1,2.1], 'k--', lw=1.0, label='Nivel de reemplazo: 2.1')
    plt.legend()
```

Resultado preliminar Calculamos el máximo del valor del TFR de todos los países y lo usamos para generar los `yticks`

[45]:
```
# Se obtiene el máximo a partir de la información original
y_maximo = TFR['Value'].max()

# Se definen los yticks usando el y_maximo
yticks = [i for i in range(0,ceil(y_maximo)+1)]
print('TFR máximo: ', y_maximo)
print('Marcas para el eje y: ', yticks)
```

```
TFR máximo:  8.8
Marcas para el eje y:  [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

[46]:
```
# Graficamos todos los países.
inicializaGrafica(y_maximo, yticks)
graficaTFR(paises)
```



¡Esta gráfica es bastante interesante y colorida! Sin embargo poco útil.

Aunque es posible identificar, si nos fijamos bien, grupos de países que inician con un TFR entre 5 y 7, que después bajan conforme pasan las décadas.

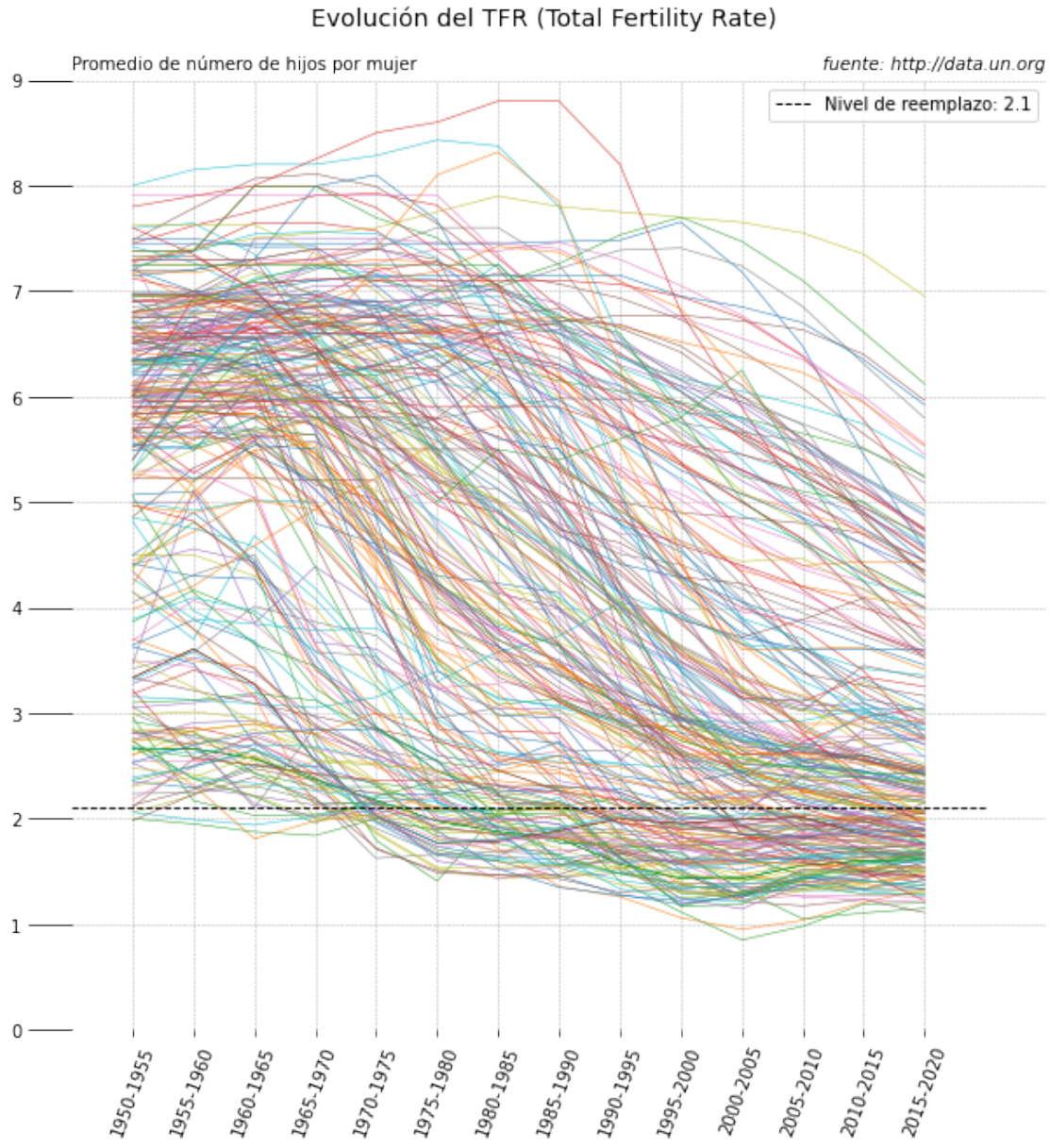
Y otro grupo de países entre el 2 y 3 que se matienen al rededor del nivel de reemplazo (2.1).

Como vimos en la celda anterior, Yemen es el país que tiene el máximo TFR, durante los años de

1985 a 1990.

Primera mejora. Haremos la misma gráfica, pero con las líneas más delgadas para intentar distinguir algo más.

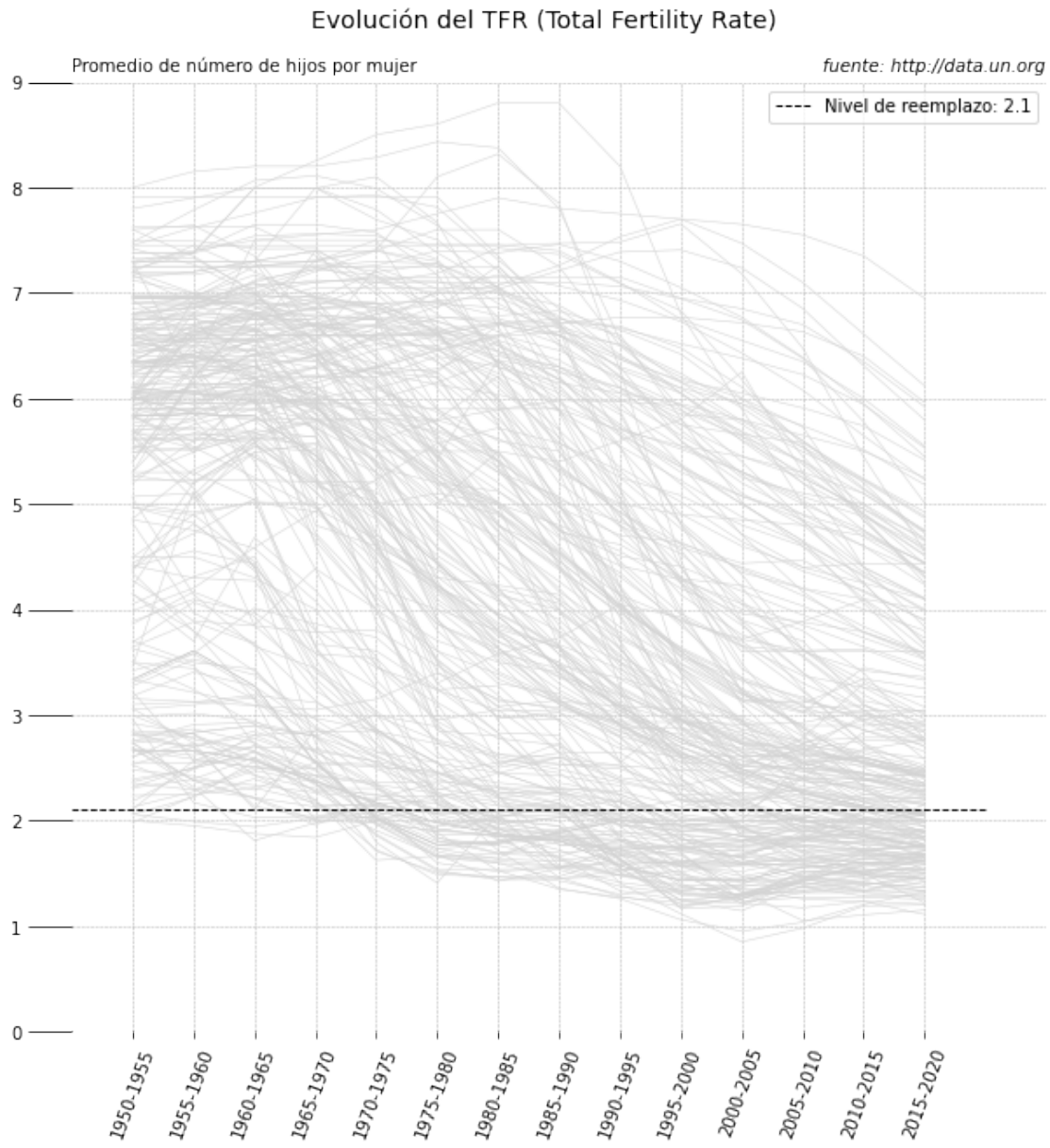
[47]:
```
inicializaGrafica(y_maximo, yticks)
graficaTFR(paises, {'lw':0.5})
```



Esta gráfica "desenreda" un poco la información, pero sigue siendo poco útil.

Segunda mejora. Usaremos un mismo color muy tenue para graficar todos los países.

[48]:
```
inicializaGrafica(y_maximo, yticks)
graficaTFR(paises, {'lw':0.5, 'c':'lightgrey'})
```



Esta gráfica es un canvas con toda la información necesaria para analizar cada país por separado. Veamos cómo.

2.10 Resultado final

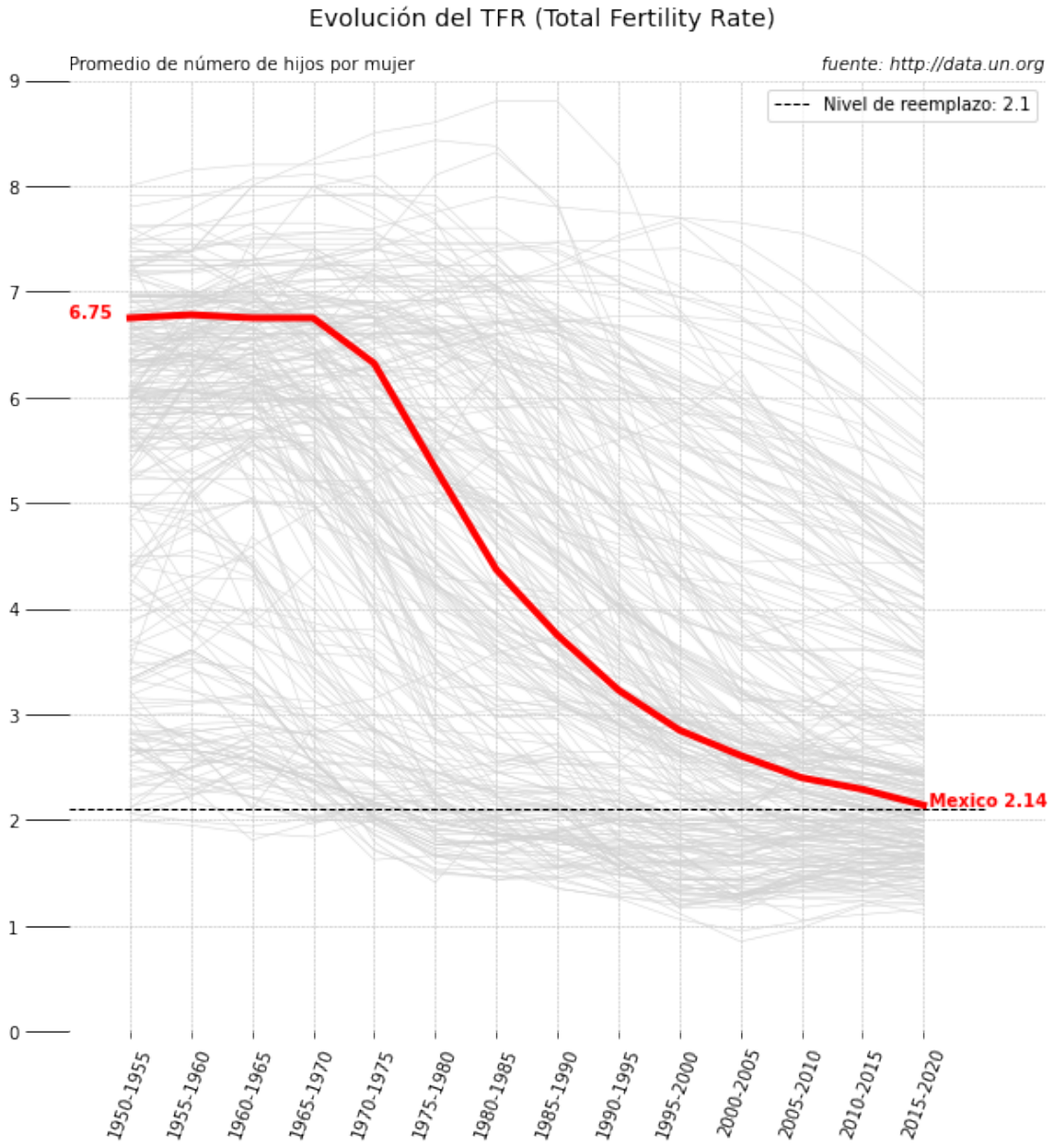
- Analizaremos los países de Suecia, España, México y Yemen, dentro de las gráficas de todos los países.

- Igualmente se van a agregar algunos otros que sean de interés.

Definimos una función para graficar un solo país con parámetros que permitan realzar la curva

[49]:
```
def graficaTFR_Pais(paises, p, parametros={}):
    """
    Realiza la gráfica de un solo país.

    Parameters
    ----------
    paises : DataFrameGroupBy
        Dataframe generado por GroupBy con la información de los países.

    parametros : dict
        Parámetros para generar la gráfica.
    """
    pais = paises.get_group(p)

    # Graficamos el país con los parámetros requeridos
    plt.plot(pais['Year(s)'], pais['Value'], **parametros)

    # Ponemos un texto al final de la curva para mostrar el
    # nombre del país y el valor final de FR
    pais_val = pais['Value'].iloc[0]
    plt.text(x = 0, y = pais_val,
             s = ' {} {:1.2f}'.format(p, pais_val),
             c = parametros['c'], weight = 'bold')

    # Ponemos el valor inicial de FR al principio de la curva.
    pais_val = pais['Value'].iloc[-1]
    plt.text(x = 14, y = pais_val,
             s = '{:1.2f} '.format(pais_val),
             c = parametros['c'], weight = 'bold')
```

[50]:
```
# Hacemos la gráfica base
inicializaGrafica(y_maximo, yticks)
graficaTFR(paises, {'lw':0.5, 'c':'lightgrey'})

# Graficamos para México con parámetros de realce (ancho 4 y color rojo)
par_mex = {'lw':4.0, 'c':'red'}
graficaTFR_Pais(paises, 'Mexico', par_mex)
```



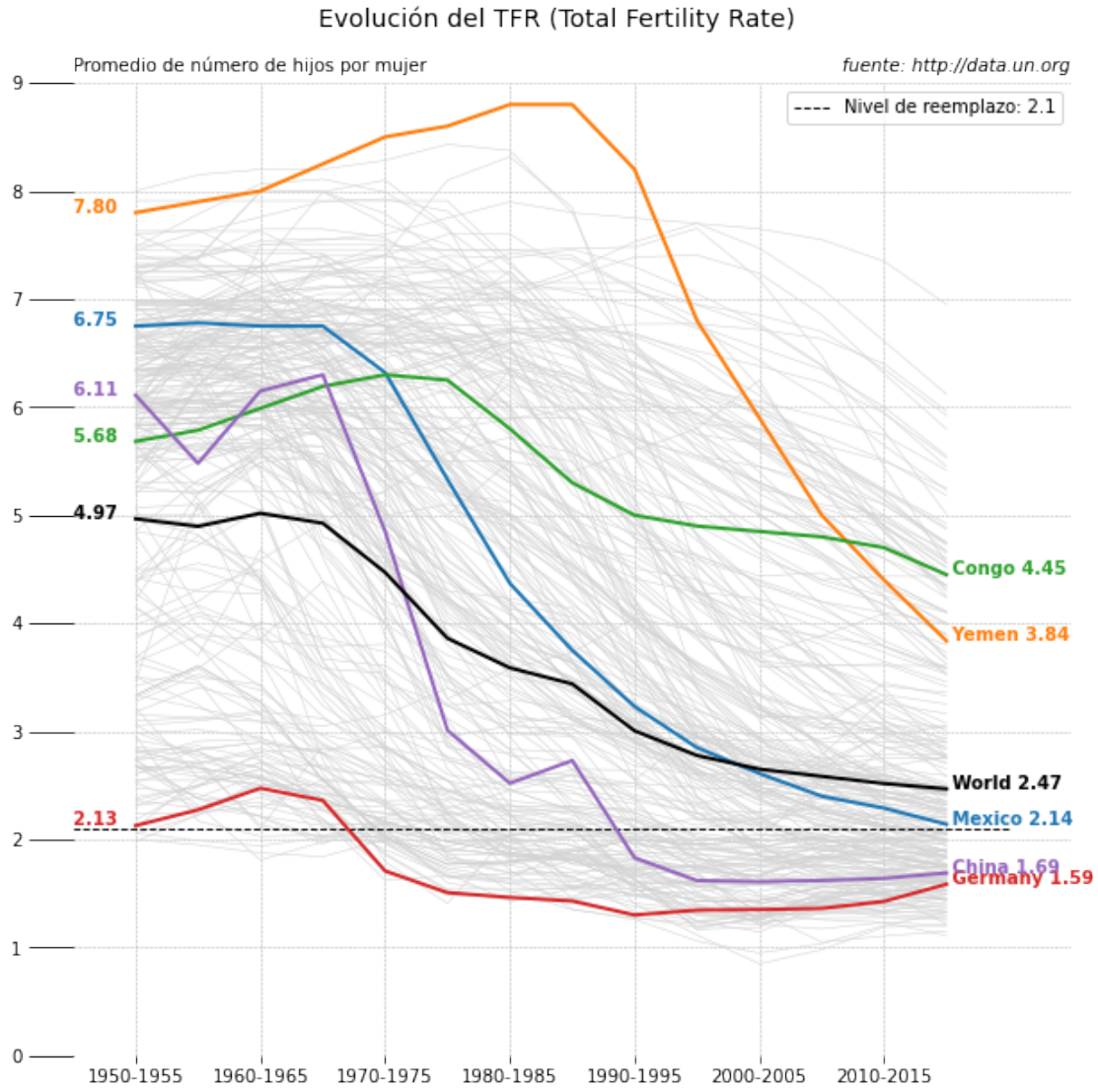
Esta gráfica nos da mucho mayor información. Además de tener todo el contexto de los demás países, podemos observar como ha cambiado el número de hijos promedio por mujer en nuestro país, desde los años 50s que tenía un valor de 6.75 y en los años 70s comenzó su declive. Fue en el año de 1974 cuando se instaló la CONAPO (Consejo Nacional de Población) y uno de sus primeros lemas fue *"La familia pequeña vive mejor"* (aún recuerdo esos promocionales de la TV). Es muy probable que este hecho haya impactado en esa disminución de los hijos por familia durante las siguientes décadas. Actualmente el valor es de 2.14, apenas un poco arriba del NR y la tendencia es a la baja. En mi experiencia, al hablar con jóvenes en edad reproductiva, parece que para un porcentaje alto de ellos (no tengo los datos exactos), ya no es un objetivo de vida tener hijos, por lo que la espectativa es que el nivel de hijos pase por abajo del NR en nuestro país en los próximos

años.

Finalmente hacemos la gráfica para varios países para hacer la comparación entre ellos.

NOTA: La base de datos también trae la información del número de hijos promedio de todo el mundo ('World') el cual graficamos para comparar.

[52]:
```
# Hacemos la gráfica base
inicializaGrafica(y_maximo, yticks)
graficaTFR(paises, {'lw':0.5, 'c':'lightgrey'})

# Definimos colores y hacemos las gráficas de los países seleccionados
colores = ['C0', 'C1', 'C2', 'C3', 'C4', 'k']

paises_l = ['Mexico', 'Yemen', 'Congo', 'Germany', 'China', 'World']
for p, c in zip(paises_l, colores):
    par = {'lw':2.0, 'c':c}
    graficaTFR_Pais(paises, p, par)

# Decidimos dibujar menos xticks para mayor claridad.
plt.xticks([13, 11, 9, 7, 5, 3, 1], rotation=0)

plt.savefig('TFR.pdf')
```



¿Puede Usted contar la historia de esta última gráfica?

¿Con este análisis es posible responder a las preguntas que hicimos al principio?

1. ¿Los países ricos están recuperando su TFR?
2. ¿Cómo es el TFR en países con un bajo PIB per cápita?
3. ¿Países en desarrollo como China y Brasil están estabilizando sus poblaciones?
4. ¿Cómo ha sido la evolución de la población en México con respecto de otros países?

2.10.1 Tarea para entrega en el Moodle.

En documento de dos páginas agregue: - Página 1 : - La gráfica del ejercicio 1 y su respuesta a la pregunta. - La gráfica del ejercicio 2 y su respuesta a la pregunta. - Página 2 : - La gráfica final del TFR que Ud. obtuvo y una descripción de la misma.

Suba su documento en formato PDF al Moodle.

2.11 Comentarios finales

En las visualizaciones que se muestran abajo se grafica el TFR y se compara con algunos factores de desarrollo para varios países de diferentes continentes y con el siguiente *GDP per capita* (en *US dollars*): - USA (63,413.5), - Alemania (46,208.4), - Japón (40,193.3), - Macao SAR (39,403.1), - Argentina (8,579.0), - Mexico (8,329.3), - Egipto (3,569.2), - Nigeria (2,097.1), - Yemen (758.1)

(Fuente: The World Data Bank).

Además, se utilizan algunas características para resaltar la visualización. Con esta información adicional ¿Ud. podría contar alguna historia?

TFR para países con diferente PIB per cápita

Algunas veces el fondo es importante (¡en otras NO!)

TFR vs PIB per cápita

TFR vs PIB per cápita vs Education

Animación del TFR de México (¡cuidado con las animaciones!)