

Prevalencia

June 4, 2020

1 Prevalencia

Vamos a analizar el influjo de la prevalencia, en el devenir de la enfermedad

En epidemiología, se denomina prevalencia a la proporción de individuos de un grupo o una población (en medicina, persona), que presentan una característica o evento determinado (en medicina, enfermedades).

Se define R_0 , como el nº de personas que infecta cada persona, en cada ciclo de infección.

Vamos a generar una lista de tablas y gráficas para ver la evolución del nº de infectados, al variar la prevalencia, y el R_0 .

```
[1]: import numpy as np
import pandas as pd
import time
from datetime import datetime, date, time, timedelta

SITUACION_INICIAL = 1
prevalencia = 0
R0 = 5.7
```

```
[2]: def Get_Header(GENERACIONES,df):
    array_fechas = []
    FECHA_INICIAL_STR = '2020-02-01'
    FECHA_INICIAL = datetime.strptime(FECHA_INICIAL_STR, "%Y-%m-%d")
    modified_date = FECHA_INICIAL
    NUM_GENERACIONES = range(1,GENERACIONES)
    for generacion in NUM_GENERACIONES:
        modified_date += timedelta(days=7)
        array_fechas.append(datetime.strptime(modified_date, "%Y-%m-%d"))
    df.columns = array_fechas
    return df

def Calcular_Cuadro_Prevalencias(R0,GENERACIONES,ARRAY_PREVALENCIAS):
    SITUACION_INICIAL=1

    diccionario_prevalencias = {}
    array=[]
```

```

for prevalencia in ARRAY_PREVALENCIAS :
    infectados_en_esta_generacion = SITUACION_INICIAL
    NUM_GENERACIONES = range(1,GENERACIONES)
    array=[]
    for generacion in NUM_GENERACIONES:
        infectados_en_esta_generacion = infectados_en_esta_generacion * R0
    ↪* (1-prevalencia)
        array.append(infectados_en_esta_generacion)
    diccionario_prevalencias['prevalencia ' + str("{:.1f}".
    ↪format(prevalencia)) + ' y R0 ' + str(R0)] = array
    df = pd.DataFrame.from_dict(diccionario_prevalencias,'index')
    df = Get_Header(GENERACIONES,df)
    df = df.astype(np.int64)
    return df.T

```

```

[3]: # Auxiliary functions
def interpolate_dataframe(df,freq):
    if freq == 'H':
        rng = pd.date_range(df.index.min(), df.index.max() + pd.Timedelta(23,
    ↪'H'), freq='H')
    elif freq == 'D' :
        rng = pd.date_range(
            datetime.strptime(str(df.index.min())[:10]+' 00:00:00', "%Y-%m-%d
    ↪%H:%M:%S"), ,
            datetime.strptime(str(df.index.max())[:10]+' 00:00:00', "%Y-%m-%d
    ↪%H:%M:%S"),
            freq='D')
        df.index = pd.to_datetime(df.index)
        df2 = df.reindex(rng)
        df = df2
        for column in df.columns :
            s = pd.Series(df[column])
            s.interpolate(method="quadratic", inplace =True)
            df[column] = pd.DataFrame([s]).T
        df.index.name = 'Fecha'
        return df

```

```

[4]: # first execution
GENERACIONES=8
ARRAY_PREVALENCIAS = np.linspace(0,0.70,8)
ARRAY_PREVALENCIAS
df = 
    ↪Calcular_Cuadro_Prevalencias(R0=R0,GENERACIONES=GENERACIONES,ARRAY_PREVALENCIAS=ARRAY_PREVA
#df = df['prevalencia 0.0 y R0 5.7']
df

```

```
[4]:
```

	prevalencia 0.0 y R0 5.7	prevalencia 0.1 y R0 5.7 \
2020-02-08	5	5
2020-02-15	32	26
2020-02-22	185	135
2020-02-29	1055	692
2020-03-07	6016	3552
2020-03-14	34296	18226
2020-03-21	195489	93502

	prevalencia 0.2 y R0 5.7	prevalencia 0.3 y R0 5.7 \
2020-02-08	4	3
2020-02-15	20	15
2020-02-22	94	63
2020-02-29	432	253
2020-03-07	1971	1011
2020-03-14	8990	4034
2020-03-21	40997	16099

	prevalencia 0.4 y R0 5.7	prevalencia 0.5 y R0 5.7 \
2020-02-08	3	2
2020-02-15	11	8
2020-02-22	40	23
2020-02-29	136	65
2020-03-07	467	188
2020-03-14	1600	535
2020-03-21	5472	1527

	prevalencia 0.6 y R0 5.7	prevalencia 0.7 y R0 5.7
2020-02-08	2	1
2020-02-15	5	2
2020-02-22	11	5
2020-02-29	27	8
2020-03-07	61	14
2020-03-14	140	25
2020-03-21	320	42

```
[5]:
```

```
from matplotlib import pyplot as plt
import pandas as pd
import numpy as np

def Get_Chart(df, title="default"):
    fig = plt.figure(figsize=(8, 6), dpi=80)
    for ca in df.columns:
        plt.plot(df[ca])
        plt.legend(df.columns)
    fig.suptitle(title, fontsize=20)
```

```
return plt
```

```
[6]: from IPython.display import display, HTML

ARRAY_ROS = [2.7,
             3.7 ,
             4.7,
             5.7,
             6.7 ]

for RO in ARRAY_ROS :

    df = □
    ↪Calcular_Cuadro_Prevalencias(RO=RO, GENERACIONES=GENERACIONES, ARRAY_PREVALENCIAS=ARRAY_PREVALENCIAS)
    display(HTML (df.to_html()))
    plt = Get_Chart(df=interpolate_dataframe(df, 'D'), title = 'Numero de □
    ↪infecciones por semana, con RO = ' + str(RO))
```

<IPython.core.display.HTML object>

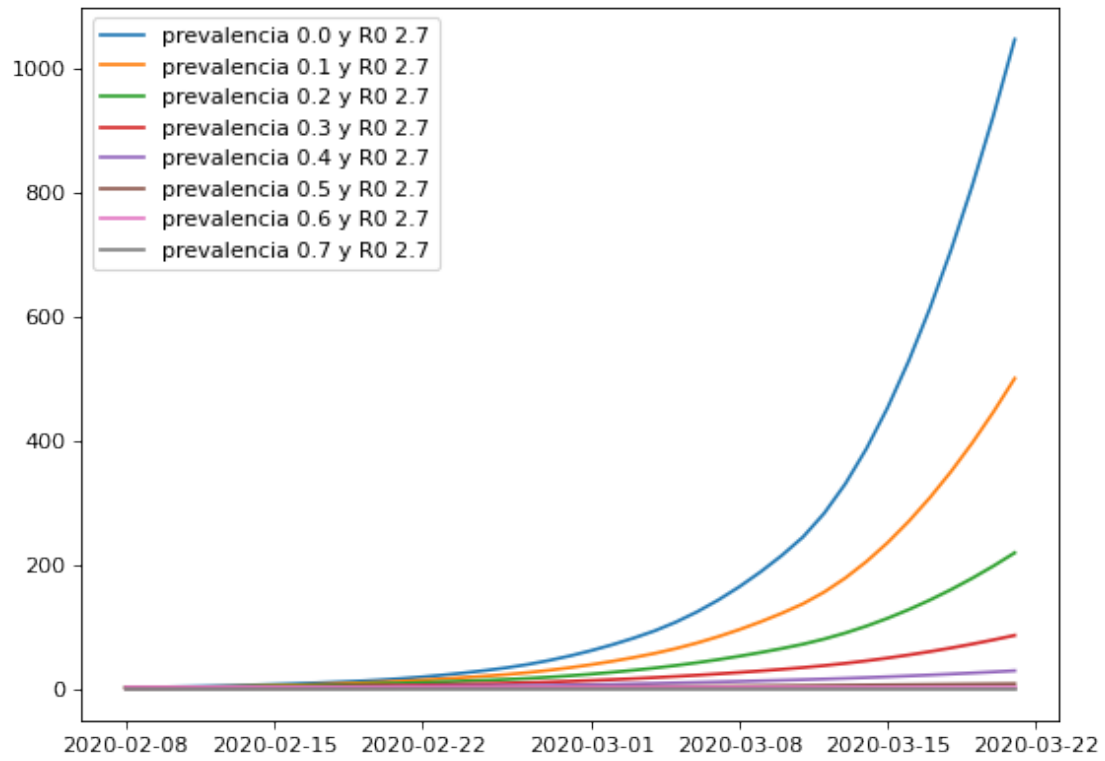
<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

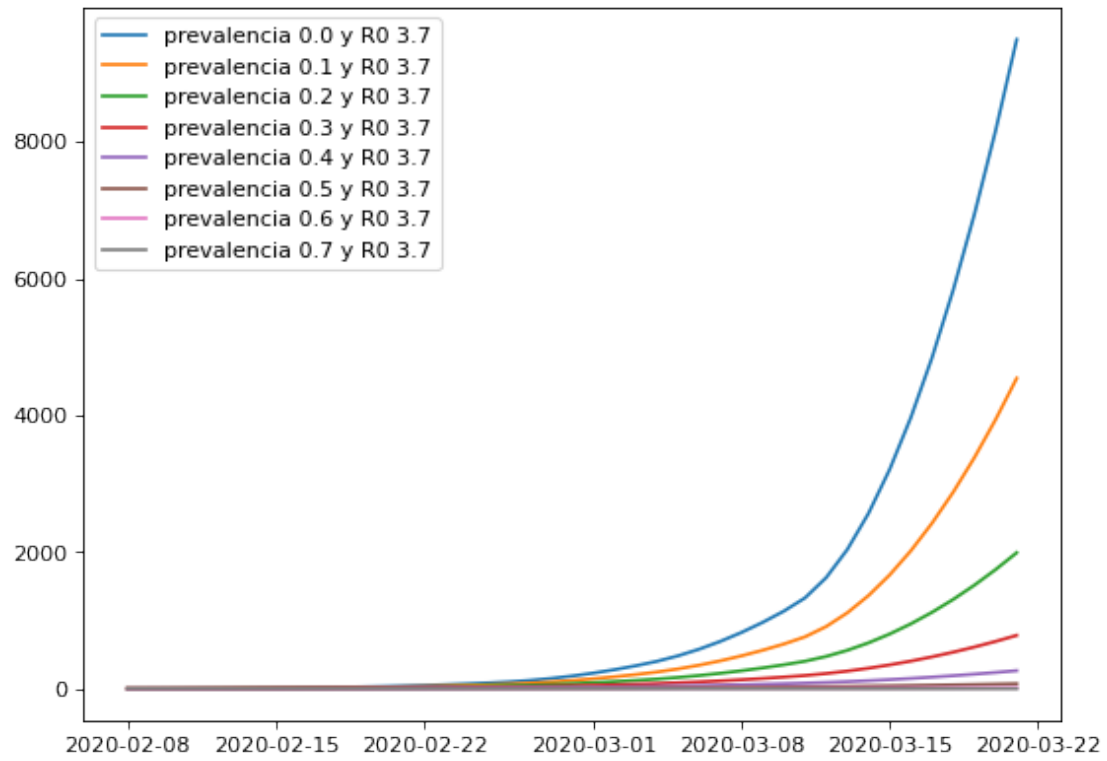
<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

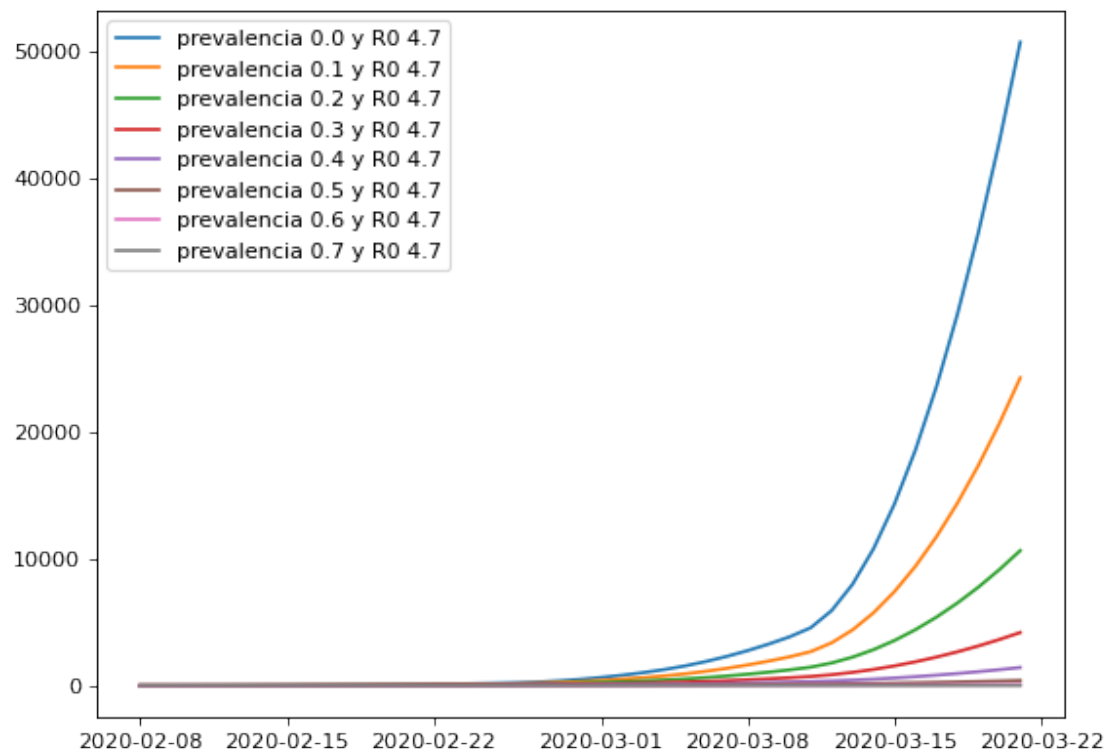
Numero de infecciones por semana, con $R_0 = 2.7$



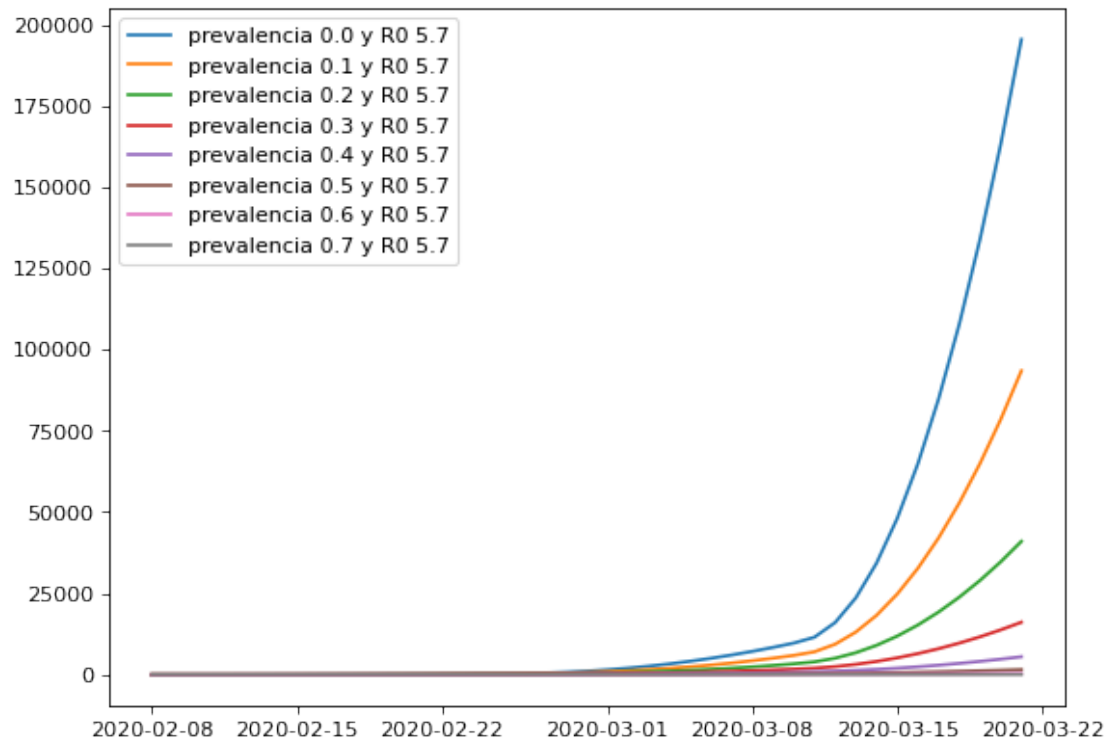
Numero de infecciones por semana, con $R_0 = 3.7$



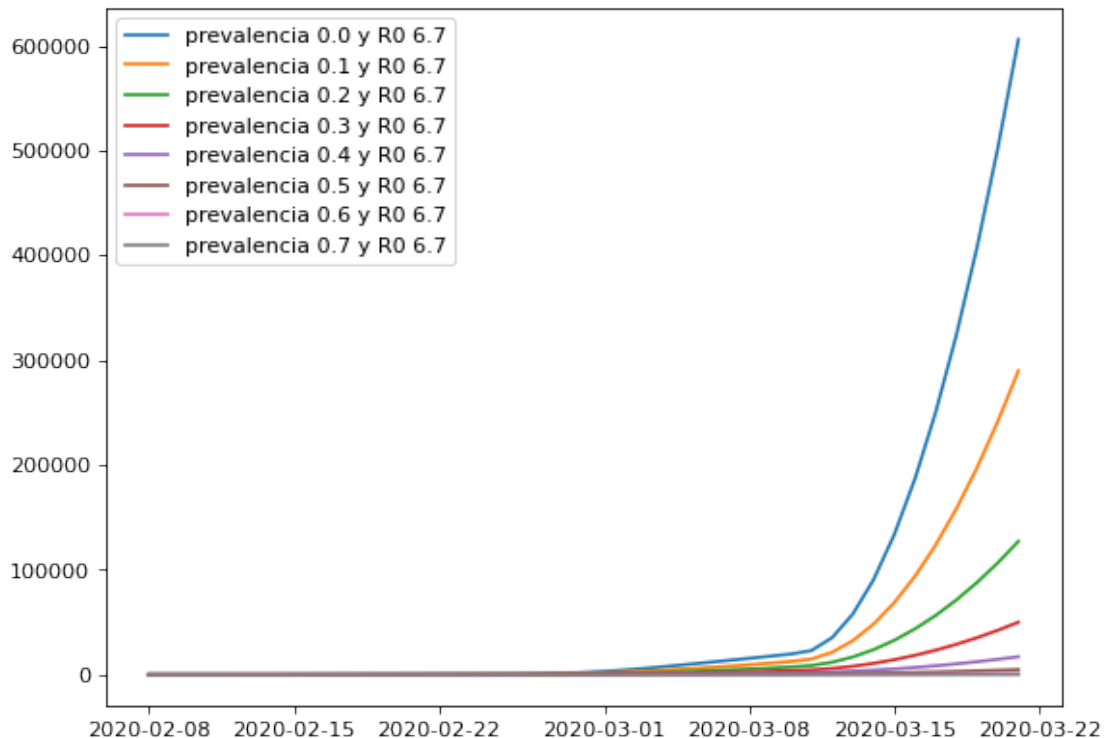
Numero de infecciones por semana, con $R_0 = 4.7$



Numero de infecciones por semana, con $R_0 = 5.7$



Numero de infecciones por semana, con $R_0 = 6.7$



TODO: - Hacer gráfico de la evolución del nº de infectados, en el confinamiento ($R_0 < 1$)

2 Conclusiones:

- Para un R_0 estimado de 5.7 del COVID-19, hace falta un 70% de prevalencia, para que no haya brotes masivos.
- Las medidas de distanciamiento social, influyen para mantener bajo el R_0 , el cual influye mucho en el número de infectados.
- Con prevalencia bajas, aunque no haya inmunidad de grupo, el nº de infectados descende significativamente. (En España, que los estudios sugieren una prevalencia menor del 10%, podría haber 1/3 ó 1/2 muertos)
- Además de la prevalencia, también es importante, tener controlado el nº de personas infectadas en cada momento:
- Hay que detectar los brotes lo mas pronto posible, y reducir el R_0 mediante la búsqueda de contactos del infectado cuando el brote es aún pequeño.
- En caso contrario, si no se puede controlar el brote, como el nº de infectados se disparará, se pueden realizar confinamientos parciales intermitentes, para reducir el R_0 por debajo de 1, hasta que el nº de infectados baje.

2.1 Bonus : ¿ Sirve de algo quedarse en casa ?

```
[7]: ARRAY_ROS = [2.7,
                 3.7 ,
                 4.7,
                 5.7,
                 6.7 ]

df_R0s = pd.DataFrame()

for MI_R0 in ARRAY_ROS :
    df = □
    ↪Calcular_Cuadro_Prevalencias(R0=MI_R0, GENERACIONES=GENERACIONES, ARRAY_PREVALENCIAS=ARRAY_PR

    df_R0s[df.columns[0]] = df[df.columns[0]]

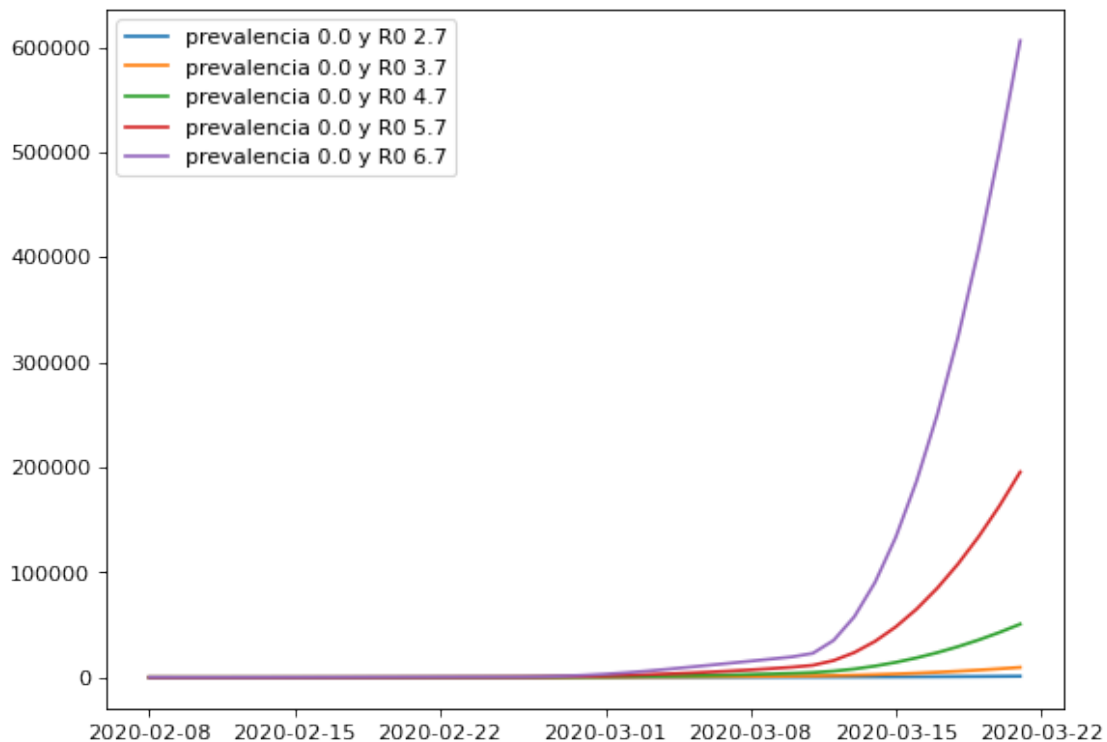
display(HTML (df_R0s.to_html()))
print("Total de infectados en cada escenario : " )
print( df_R0s.astype(np.int64).sum(axis=0) )

plt = Get_Chart(interpolate_dataframe(df_R0s,'D') ,title= 'Comparativa de nº de □
↪infectados variando el R0')
```

<IPython.core.display.HTML object>

```
Total de infectados en cada escenario :
prevalencia 0.0 y R0 2.7      1657
prevalencia 0.0 y R0 3.7     13004
prevalencia 0.0 y R0 4.7     64350
prevalencia 0.0 y R0 5.7    237078
prevalencia 0.0 y R0 6.7    712395
dtype: int64
```

Comparativa de nº de infectados variando el R0



```
[8]: #df = interpolate_dataframe(df_R0s)
```

Como se ve en la gráfica, bajar el R0, es muy importante, para detener el nº de infectados, y por ende en número de los fallecidos.

2.2 Re- Bonus : Se me ha hecho larguísimo, ¿ Podría haber estado menos tiempo en casa ?

```
[9]: def comparacion_semanas():  
    GENERACIONES=8  
    SITUACION_INICIAL=1  
    prevalencia=0  
    diccionario_prevalencias={}  
    R0=5.7  
    i=1  
    for NUM_GENERACIONES in range(5,8) :  
        infectados_en_esta_generacion=SITUACION_INICIAL  
        array = []  
        for generacion in range(1,NUM_GENERACIONES):
```

```

        infectados_en_esta_generacion = infectados_en_esta_generacion * R0
    ↪* (1-prevalencia)
        array.append(infectados_en_esta_generacion)
        valor_actual = array[-1]
        # Calculamos el R0 con el confinamiento
        # En alemania confinada el R0 estimado es un 0.7. Nosotros debería ser
    ↪mas bajo
        NUEVO_R0 = 0.5
        while infectados_en_esta_generacion > 1 :
            infectados_en_esta_generacion = infectados_en_esta_generacion *
    ↪NUEVO_R0 * (1-prevalencia)
            array.append(infectados_en_esta_generacion)
            diccionario_prevalencias[' R0 ' + str(R0) + ', parando en la semana ' +
    ↪str(generacion)] = array
            i=i+1

        df = pd.DataFrame.from_dict(diccionario_prevalencias,'index')
        df = Get_Header(df.shape[1]+1,df)
        df = df.T
        df.index = pd.to_datetime(df.index)

    return df

```

```

[10]: df = comparacion_semanas()
      df= df.fillna(0)

      print("Total de infectados en cada escenario : " )
      print( df.astype(np.int64).sum(axis=0) )

      plt = Get_Chart(df = interpolate_dataframe(df=df,freq='H'),title= 'Comparativa
    ↪de infectados y tiempo de confinamiento, según el momento de empezar')
      df.astype(int)

```

Total de infectados en cada escenario :

R0 5.7, parando en la semana 4	2326
R0 5.7, parando en la semana 5	13304
R0 5.7, parando en la semana 6	75877

dtype: int64

```

[10]:
      R0 5.7, parando en la semana 4    R0 5.7, parando en la semana 5 \
2020-02-08                             5                             5
2020-02-15                             32                             32
2020-02-22                             185                            185
2020-02-29                            1055                           1055
2020-03-07                             527                           6016
2020-03-14                             263                           3008

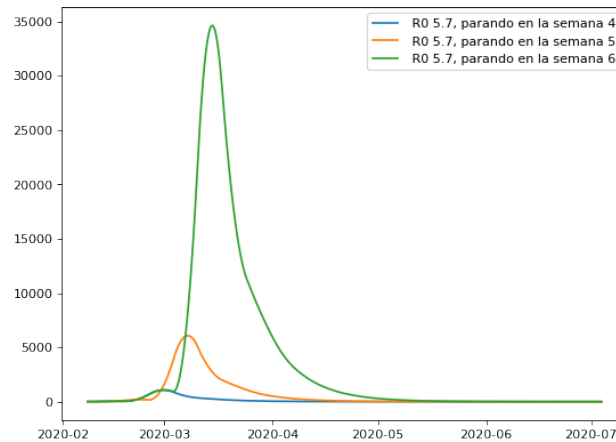
```

2020-03-21	131	1504
2020-03-28	65	752
2020-04-04	32	376
2020-04-11	16	188
2020-04-18	8	94
2020-04-25	4	47
2020-05-02	2	23
2020-05-09	1	11
2020-05-16	0	5
2020-05-23	0	2
2020-05-30	0	1
2020-06-06	0	0
2020-06-13	0	0
2020-06-20	0	0
2020-06-27	0	0
2020-07-04	0	0

R0 5.7, parando en la semana 6

2020-02-08	5
2020-02-15	32
2020-02-22	185
2020-02-29	1055
2020-03-07	6016
2020-03-14	34296
2020-03-21	17148
2020-03-28	8574
2020-04-04	4287
2020-04-11	2143
2020-04-18	1071
2020-04-25	535
2020-05-02	267
2020-05-09	133
2020-05-16	66
2020-05-23	33
2020-05-30	16
2020-06-06	8
2020-06-13	4
2020-06-20	2
2020-06-27	1
2020-07-04	0

Comparativa de infectados y tiempo de confinamiento, según el momento de empezar



Como se ve en el anterior gráfico, el tiempo que dura confinamiento, y el nº de infectados varía enormemente.

Sin duda la demora en adoptar las restricciones - como se ve con esta gráfica de datos teóricos- ha influido en el tiempo de confinamiento, y mucho peor, ha costado un gran número de vidas.

[]: