

Madrid_Pain_Graphs

October 13, 2020

1 Informes de la comunidad de Madrid

Actualizado diariamente, este documento se [visualiza mejor aquí](#).

Datos de la situación de la infección por coronavirus en la Comunidad de Madrid.

Nos descargamos los datos, agrupamos, y calculamos :

- Gráfico de seguimiento.
- Muertes medias diarias, últimos 7 días.
- Muertes medias diarias desde que la comunidad de Madrid publica datos.

```
[1]: # Miramos si hay nuevos datos a descargar.

!# cd ../data/; FILELIST=" 200509 200508 200507 200506 200505 200504 200503_
→200502 200501 200430 200429 200428 200427 200426 200425 200424 200423 200422_
→200510 200511 200512 200513 200514 200515 200516 200517 200518 200519 200520_
→200521 200522 200523 200524 200525 200526 200527 200528 200529 200530 200609_
→200608 200607 200606 200605 200604 200603 200602 200601 200610 200611 200612_
→200613 200614 200615 200616 200617 200618 200619 200620 200621 200622 200623_
→200624 200625 200626 200627 200628 200629 200630 " ; for fecha in `echo_
→$FILELIST` ; do FILE=${fecha}_cam_covid19.pdf ; [ ! -f ../data/${FILE} ] _
→&& echo $FILE::::: && wget https://www.comunidad.madrid/sites/default/
→files/doc/sanidad/${FILE} 1>/dev/null 2>/dev/null && ls -altr $FILE ; done

# Miramos solo hoy y los ultimos diez dias
! cd ../data/; FILELIST=`seq -w 0 8 | while read i ; do date +%y%m%d -d "$i day_
→ago" ; done` ; for fecha in `echo $FILELIST` ; do _
→FILE=${fecha}_cam_covid19.pdf ; [ ! -f ../data/${FILE} ] && echo $FILE::::: _
→&& wget https://www.comunidad.madrid/sites/default/files/doc/sanidad/
→$FILE 1>/dev/null 2>/dev/null && ls -altr $FILE ; done

! cd ../data/; FILELIST=`seq -w 0 8 | while read i ; do date +%y%m%d -d "$i day_
→ago" ; done` ; for fecha in `echo $FILELIST` ; do FILE=${fecha}_cam_covid19.
→pdf ; [ ! -f ../data/${FILE} ] && echo $FILE::::: && wget https://www.
→comunidad.madrid/sites/default/files/doc/sanidad/${FILE} 1>/dev/null 2>/dev/
→null && ls -altr $FILE ; done
```

```
! cd ../data/; FILELIST=`seq -w 0 8 | while read i ; do date +%Y%m%d -d "$i day_
→ago" ; done` ; for fecha in `echo $FILELIST` ; do _
→FILE=${fecha}_cam_covid19.pdf ; [ ! -f ../data/${FILE} ] && echo $FILE:~~~~:
→ && wget https://www.comunidad.madrid/sites/default/files/doc/sanidad/
→$FILE 1>/dev/null 2>/dev/null && ls -altr $FILE ; done
! cd ../data/; FILELIST=`seq -w 0 8 | while read i ; do date +%Y%m%d -d "$i day_
→ago" ; done` ; for fecha in `echo $FILELIST` ; do _
→FILE=${fecha}_cam_covid19.pdf ; [ ! -f ../data/${FILE} ] && echo $FILE:~~~~:
→ && wget https://www.comunidad.madrid/sites/default/files/$FILE 1>/dev/
→null 2>/dev/null && ls -altr $FILE ; done
#200902_cam_covid19.pdf
```

```
201012_cam_covid19.pdf:~~~~:
201011_cam_covid19.pdf:~~~~:
201010_cam_covid19.pdf:~~~~:
201013cam_covid19.pdf:~~~~:
201012cam_covid19.pdf:~~~~:
201011cam_covid19.pdf:~~~~:
201010cam_covid19.pdf:~~~~:
201009cam_covid19.pdf:~~~~:
201008cam_covid19.pdf:~~~~:
201007cam_covid19.pdf:~~~~:
201006cam_covid19.pdf:~~~~:
201005cam_covid19.pdf:~~~~:
20201013_cam_covid19.pdf:~~~~:
20201012_cam_covid19.pdf:~~~~:
20201011_cam_covid19.pdf:~~~~:
20201010_cam_covid19.pdf:~~~~:
20201009_cam_covid19.pdf:~~~~:
20201008_cam_covid19.pdf:~~~~:
20201007_cam_covid19.pdf:~~~~:
20201006_cam_covid19.pdf:~~~~:
20201005_cam_covid19.pdf:~~~~:
201012_cam_covid19.pdf:~~~~:
201011_cam_covid19.pdf:~~~~:
201010_cam_covid19.pdf:~~~~:
```

```
[2]: from tabula import read_pdf
from IPython.display import display, HTML
import os
import pandas as pd
import glob
import re
from tqdm.notebook import tqdm
import warnings
import os.path
```

```
warnings.filterwarnings('ignore')

os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.141-1.b16.
↳el7_3.x86_64/jre"

# Auxiliary functions
from datetime import datetime, date, time, timedelta

""" Rellenar dias vacios con interpolacion """
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
    return df

def get_daily_date_new_format(fecha):

    file_path = '../data/'+fecha+'_cam_covid19.pdf'
    if not os.path.isfile(file_path):
        file_path = '../data/'+fecha+'cam_covid19.pdf'
    #print("Analizando:" + file_path)
    df_pdf = read_pdf(file_path,area=(000, 600, 400, 800) , pages='1')

    #print("1 get_daily_date_new_format ",fecha)

    df = df_pdf[0]
    df = df['Unnamed: 0'].astype(str).str.replace(r".", '').replace("(", ' ')
    df = df.T
    df.columns = df.iloc[0]
    df = df.iloc[1:]

    #print("2 get_daily_date_new_format")
```

```

df = pd.DataFrame(data=df)
df

dict = {}
try:
    df2_pdf = read_pdf(file_path, area=(300, 100, 800, 400) , pages='1')
    dict['PACIENTES UCI DIA'] = df2_pdf[0].loc[3:3].values[0][1].
    ↪replace(".", '')
    dict['PACIENTES UCI ACUMULADOS'] = df2_pdf[0].loc[6:6].values[0][1].
    ↪replace(".", '')
except Exception as e:
    print(f"{fecha} mal parseada: {e}")

dict['HOSPITALES'] = df[df['Unnamed: 0'].str.contains('Hospitales')].
    ↪iloc[0]['Unnamed: 0'].split(' ')[0]
dict['DOMICILIOS'] = df[df['Unnamed: 0'].str.contains('Domicilios')].
    ↪iloc[0]['Unnamed: 0'].split(' ')[0]
dict['CENTROS SOCIO SANITARIOS'] = df[df['Unnamed: 0'].str.
    ↪contains('Centros')].iloc[0]['Unnamed: 0'].split(' ')[0]
dict['OTROS LUGARES'] = df[df['Unnamed: 0'].str.contains('otros')].
    ↪iloc[0]['Unnamed: 0'].split(' ')[0]
#print("3 get_daily_date_new_format")

cadena_a_parsear = df[df['Unnamed: 0'].str.contains('otal')].
    ↪iloc[0]['Unnamed: 0']

dict['FALLECIDOS TOTALES'] = re.search(r'(\d+)', cadena_a_parsear)[0]

#print("4 get_daily_date_new_format")
df = pd.DataFrame.from_dict(dict, orient='index').T
#print("4.5 get_daily_date_new_format")

try:
    df['Fecha'] = pd.to_datetime(fecha, format='%y%m%d')
except :
    df['Fecha'] = pd.to_datetime(fecha, format='%Y%m%d')

#print("5 get_daily_date_new_format")

df.set_index('Fecha', inplace=True, drop=True)
#print(df)
return df

def get_daily_data(fecha):

```

```

#print(f"""get_daily_data: {fecha}""")
#print(f"""../data/{fecha}_cam_covid19.pdf""")

if fecha > '200512' :
    return get_daily_date_new_format(fecha)

col2str = {'dtype': str}
kwargs = {'output_format': 'dataframe',
          'pandas_options': col2str,
          'stream': True}

df_pdf = read_pdf('../data/'+fecha+'_cam_covid19.
→pdf',pages='1',multiple_tables = True,**kwargs)

df = df_pdf[0]

df = df[df['Unnamed: 0'].notna()]
df = df[(df['Unnamed: 0']=='HOSPITALES') | (df['Unnamed: 0'] ==
→'DOMICILIOS') | (df['Unnamed: 0'] == 'CENTROS SOCIO SANITARIOS') |
→(df['Unnamed: 0'] == 'OTROS LUGARES') | (df['Unnamed: 0'] == 'FALLECIDOS_
→TOTALES')]]
df = df[['Unnamed: 0','Unnamed: 2']]
df['Unnamed: 2'] = df['Unnamed: 2'].astype(str).str.replace(r".", '')
df = df.T
df.columns = df.iloc[0]
df = df.iloc[1:]

df['Fecha'] = pd.to_datetime(fecha, format='%y%m%d')
df = df.rename_axis(None)

df.set_index('Fecha', inplace=True, drop=True)
df.index
df.dropna()
#df = df.T
return df

def get_all_data( ):
    #BLACKLIST = ["200429","200422"]
    #BLACKLIST = ["200514",]
    BLACKLIST = []
    df = pd.DataFrame()
    list_df = []

    pdf_list= sorted(glob.glob('../data/*_covid19.pdf'),
                     key=os.path.getmtime,
                     reverse=True )

```

```

for pdf_file in tqdm(pdf_list,
                      desc="Procesando pdfs diarios"):
    # extract fecha from username , eg : ../data/2200422_cam_covid19.pdf
    fecha = pdf_file.split('/')[2].split('_')[0].replace('cam_', '').
    ↪replace('_cam_', '').replace('cam', '')
    if fecha not in BLACKLIST:
        #print("processing", fecha)
        df = get_daily_data(fecha)
        list_df.append(df)

df = pd.concat(list_df)
df = df.fillna(0)
df = df.astype(int)
df = df.drop_duplicates()

df = df.sort_values(by=['Fecha'], ascending=True)

df['HOSPITALES hoy'] = df['HOSPITALES'] - df['HOSPITALES'].shift(1)
df['CENTROS SOCIOSANITARIOS hoy'] = df['CENTROS SOCIOSANITARIOS'] -
    ↪df['CENTROS SOCIOSANITARIOS'].shift(1)
df['FALLECIDOS TOTALES hoy'] = df['FALLECIDOS TOTALES'] - df['FALLECIDOS_
    ↪TOTALES'].shift(1)

df = df.sort_values(by=['Fecha'], ascending=False)

return df

total = get_all_data()

total.to_csv('/root/kaggle/covid19-madrid/madrid_results.csv')
total

```

HBox(children=(FloatProgress(value=0.0, description='Procesando pdfs diarios', max=144.0, style=

Got stderr: oct 13, 2020 7:50:10 PM
 org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
 ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:12 PM
 org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
 ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:13 PM
 org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
 ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:15 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:16 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:18 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:19 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:21 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:22 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:24 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:25 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:27 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:28 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:29 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:31 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:33 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:35 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:36 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:38 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

Got stderr: oct 13, 2020 7:50:39 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'

200519 mal parseada: index 1 is out of bounds for axis 0 with size 1
200518 mal parseada: index 1 is out of bounds for axis 0 with size 1
200517 mal parseada: index 1 is out of bounds for axis 0 with size 1
200516 mal parseada: index 1 is out of bounds for axis 0 with size 1
200515 mal parseada: index 1 is out of bounds for axis 0 with size 1
200514 mal parseada: index 1 is out of bounds for axis 0 with size 1
200513 mal parseada: index 1 is out of bounds for axis 0 with size 1

Got stderr: oct 13, 2020 7:55:49 PM
org.apache.pdfbox.pdmodel.font.PDCIDFontType2 <init>
INFORMACIÓN: OpenType Layout tables used in font CIDFont+F1 are not implemented
in PDFBox and will be ignored
oct 13, 2020 7:55:49 PM org.apache.pdfbox.pdmodel.font.PDCIDFontType2 <init>
INFORMACIÓN: OpenType Layout tables used in font CIDFont+F2 are not implemented
in PDFBox and will be ignored
oct 13, 2020 7:55:49 PM org.apache.pdfbox.pdmodel.font.PDCIDFontType2 <init>
INFORMACIÓN: OpenType Layout tables used in font CIDFont+F3 are not implemented
in PDFBox and will be ignored
oct 13, 2020 7:55:50 PM org.apache.pdfbox.pdmodel.font.PDCIDFontType2 <init>
INFORMACIÓN: OpenType Layout tables used in font CIDFont+F1 are not implemented
in PDFBox and will be ignored
oct 13, 2020 7:55:50 PM org.apache.pdfbox.pdmodel.font.PDCIDFontType2 <init>
INFORMACIÓN: OpenType Layout tables used in font CIDFont+F2 are not implemented
in PDFBox and will be ignored
oct 13, 2020 7:55:50 PM org.apache.pdfbox.pdmodel.font.PDCIDFontType2 <init>
INFORMACIÓN: OpenType Layout tables used in font CIDFont+F3 are not implemented
in PDFBox and will be ignored

oct 13, 2020 7:55:50 PM org.apache.pdfbox.pdmodel.font.PDCIDFontType2 <init>
 INFORMACIÓN: OpenType Layout tables used in font CIDFont+F1 are not implemented
 in PDFBox and will be ignored
 oct 13, 2020 7:55:50 PM org.apache.pdfbox.pdmodel.font.PDCIDFontType2 <init>
 INFORMACIÓN: OpenType Layout tables used in font CIDFont+F2 are not implemented
 in PDFBox and will be ignored
 oct 13, 2020 7:55:50 PM org.apache.pdfbox.pdmodel.font.PDCIDFontType2 <init>
 INFORMACIÓN: OpenType Layout tables used in font CIDFont+F3 are not implemented
 in PDFBox and will be ignored

[2]: CENTROS SOCIOSANITARIOS DOMICILIOS FALLECIDOS TOTALES \

Fecha			
2020-10-13	4883	1047	17038
2020-10-09	4882	1046	16999
2020-10-08	4878	1042	16906
2020-10-07	4878	1042	16906
2020-10-06	4874	1035	16742
...
2020-04-26	4236	798	12855
2020-04-25	4170	788	12612
2020-04-24	4068	775	12252
2020-04-23	3996	769	12056
2020-04-22	3932	761	11852

	HOSPITALES	OTROS LUGARES	PACIENTES UCI ACUMULADOS	\
Fecha				
2020-10-13	11078	30	5445	
2020-10-09	11041	30	5354	
2020-10-08	10956	30	5323	
2020-10-07	10956	30	5288	
2020-10-06	10803	30	5262	
...	
2020-04-26	7800	21	0	
2020-04-25	7633	21	0	
2020-04-24	7388	21	0	
2020-04-23	7271	20	0	
2020-04-22	7144	15	0	

	PACIENTES UCI DIA	HOSPITALES hoy	CENTROS SOCIOSANITARIOS hoy	\
Fecha				
2020-10-13	473	37.0	1.0	
2020-10-09	487	85.0	4.0	
2020-10-08	487	0.0	0.0	
2020-10-07	487	153.0	4.0	
2020-10-06	489	0.0	0.0	

...
2020-04-26	0	167.0	66.0
2020-04-25	0	245.0	102.0
2020-04-24	0	117.0	72.0
2020-04-23	0	127.0	64.0
2020-04-22	0	NaN	NaN

FALLECIDOS TOTALES hoy

Fecha	
2020-10-13	39.0
2020-10-09	93.0
2020-10-08	0.0
2020-10-07	164.0
2020-10-06	0.0
...	...
2020-04-26	243.0
2020-04-25	360.0
2020-04-24	196.0
2020-04-23	204.0
2020-04-22	NaN

[131 rows x 10 columns]

```
[3]: interpolate_dataframe(total, 'D')
```

[3]:	CENTROS SOCIOSSANITARIOS	DOMICILIOS	FALLECIDOS TOTALES	\
	2020-04-22	3932.000000	761.000000	11852.000000
	2020-04-23	3996.000000	769.000000	12056.000000
	2020-04-24	4068.000000	775.000000	12252.000000
	2020-04-25	4170.000000	788.000000	12612.000000
	2020-04-26	4236.000000	798.000000	12855.000000
	
	2020-10-09	4882.000000	1046.000000	16999.000000
	2020-10-10	4885.164748	1049.230540	17075.568484
	2020-10-11	4886.386331	1050.474053	17107.591312
	2020-10-12	4885.664748	1049.730540	17095.068484
	2020-10-13	4883.000000	1047.000000	17038.000000

	HOSPITALES	OTROS LUGARES	PACIENTES UCI	ACUMULADOS	\
	2020-04-22	7144.000000	15.0	0.000000	
	2020-04-23	7271.000000	20.0	0.000000	
	2020-04-24	7388.000000	21.0	0.000000	
	2020-04-25	7633.000000	21.0	0.000000	
	2020-04-26	7800.000000	21.0	0.000000	
	
	2020-10-09	11041.000000	30.0	5354.000000	
	2020-10-10	11111.173190	30.0	5381.394116	

2020-10-11	11140.730921	30.0	5405.692154
2020-10-12	11129.673190	30.0	5426.894116
2020-10-13	11078.000000	30.0	5445.000000

	PACIENTES UCI DIA	HOSPITALES hoy	CENTROS SOCIOSSANITARIOS hoy \
2020-04-22	0.000000	NaN	NaN
2020-04-23	0.000000	127.000000	64.000000
2020-04-24	0.000000	117.000000	72.000000
2020-04-25	0.000000	245.000000	102.000000
2020-04-26	0.000000	167.000000	66.000000
...
2020-10-09	487.000000	85.000000	4.000000
2020-10-10	485.726917	167.192902	7.313930
2020-10-11	482.969223	186.590536	7.918573
2020-10-12	478.726917	143.192902	5.813930
2020-10-13	473.000000	37.000000	1.000000

	FALLECIDOS TOTALES hoy
2020-04-22	NaN
2020-04-23	204.000000
2020-04-24	196.000000
2020-04-25	360.000000
2020-04-26	243.000000
...	...
2020-10-09	93.000000
2020-10-10	182.280755
2020-10-11	203.041006
2020-10-12	155.280755
2020-10-13	39.000000

[175 rows x 10 columns]

```
[4]: total
df = total
df = df.fillna(0)
df = df.astype(int)
df
```

	CENTROS SOCIOSSANITARIOS	DOMICILIOS	FALLECIDOS TOTALES \
Fecha			
2020-10-13	4883	1047	17038
2020-10-09	4882	1046	16999
2020-10-08	4878	1042	16906
2020-10-07	4878	1042	16906
2020-10-06	4874	1035	16742
...
2020-04-26	4236	798	12855

2020-04-25	4170	788	12612
2020-04-24	4068	775	12252
2020-04-23	3996	769	12056
2020-04-22	3932	761	11852

Fecha	HOSPITALES	OTROS LUGARES	PACIENTES UCI	ACUMULADOS \
2020-10-13	11078	30		5445
2020-10-09	11041	30		5354
2020-10-08	10956	30		5323
2020-10-07	10956	30		5288
2020-10-06	10803	30		5262
...
2020-04-26	7800	21		0
2020-04-25	7633	21		0
2020-04-24	7388	21		0
2020-04-23	7271	20		0
2020-04-22	7144	15		0

Fecha	PACIENTES UCI	DIA	HOSPITALES hoy	CENTROS SOCIO SANITARIOS hoy \
2020-10-13		473	37	1
2020-10-09		487	85	4
2020-10-08		487	0	0
2020-10-07		487	153	4
2020-10-06		489	0	0
...
2020-04-26		0	167	66
2020-04-25		0	245	102
2020-04-24		0	117	72
2020-04-23		0	127	64
2020-04-22		0	0	0

Fecha	FALLECIDOS TOTALES hoy
2020-10-13	39
2020-10-09	93
2020-10-08	0
2020-10-07	164
2020-10-06	0
...	...
2020-04-26	243
2020-04-25	360
2020-04-24	196
2020-04-23	204
2020-04-22	0

[131 rows x 10 columns]

```
[5]: total
VENTANA_MEDIA_MOVIL=7
df = interpolate_dataframe(total,'D')
df.index.name = 'Fecha'
df = df.sort_values(by=['Fecha'], ascending=True)
df['HOSPITALES hoy'] = df['HOSPITALES'] - df['HOSPITALES'].shift(1)
df['CENTROS SOCIO SANITARIOS hoy'] = df['CENTROS SOCIO SANITARIOS'] - df['CENTROS_
↳SOCIO SANITARIOS'].shift(1)
df['FALLECIDOS TOTALES hoy'] = df['FALLECIDOS TOTALES'] - df['FALLECIDOS_
↳TOTALES'].shift(1)

df['MA CENTROS SOCIO SANITARIOS hoy'] = df['CENTROS SOCIO SANITARIOS hoy'].
↳rolling(window=VENTANA_MEDIA_MOVIL).mean()
df['MA HOSPITALES hoy'] = df['HOSPITALES hoy'].
↳rolling(window=VENTANA_MEDIA_MOVIL).mean()
df['MA FALLECIDOS TOTALES hoy'] = df['FALLECIDOS TOTALES hoy'].
↳rolling(window=VENTANA_MEDIA_MOVIL).mean()

df = df.sort_index(ascending=False)
df_master = df.copy()
```

```
[6]: total.head()
```

```
[6]:
```

	CENTROS SOCIO SANITARIOS	DOMICILIOS	FALLECIDOS TOTALES	\
Fecha				
2020-10-13	4883	1047	17038	
2020-10-09	4882	1046	16999	
2020-10-08	4878	1042	16906	
2020-10-07	4878	1042	16906	
2020-10-06	4874	1035	16742	

	HOSPITALES	OTROS LUGARES	PACIENTES UCI ACUMULADOS	\
Fecha				
2020-10-13	11078	30	5445	
2020-10-09	11041	30	5354	
2020-10-08	10956	30	5323	
2020-10-07	10956	30	5288	
2020-10-06	10803	30	5262	

	PACIENTES UCI DIA	HOSPITALES hoy	CENTROS SOCIO SANITARIOS hoy	\
Fecha				
2020-10-13	473	37.0	1.0	
2020-10-09	487	85.0	4.0	
2020-10-08	487	0.0	0.0	

2020-10-07	487	153.0	4.0
2020-10-06	489	0.0	0.0

FALLECIDOS TOTALES hoy	
Fecha	
2020-10-13	39.0
2020-10-09	93.0
2020-10-08	0.0
2020-10-07	164.0
2020-10-06	0.0

```
[7]: # Hacemos lo contrario
# En lugar de sacar el nº de muertos dado el nº de infectados, como lo primero
# lo sabemos (en madrid), sacamos lo segundo y extrapolamos al conjunto de
# España
df = df_master

R0_estimada = df['FALLECIDOS TOTALES hoy'].values[0:7].sum() / df['FALLECIDOS_
# TOTALES hoy'].values[7:14].sum()
print(df['FALLECIDOS TOTALES hoy'].values[0:7].sum(), df['FALLECIDOS TOTALES_
# hoy'].values[7:14].sum() )
print(f"R0_estimada = {R0_estimada}")
PROPORCION_ENFERMOS_MUERTOS=750000/15000 # Esta es la proporcion enfermos
# muertos (15.000 muertos para 750.000 afectados)
RATIO_NO_HEMOS_COLAPSADO=2 # La mitad de los muertos se ha calculado del
# colapso. Como ahora no hemos colapsado
PESO_MADRID_MUERTES_TOTALES=1/3
casos_españa_estimados = df['FALLECIDOS TOTALES hoy'].values[0:5].sum() *
# PROPORCION_ENFERMOS_MUERTOS * RATIO_NO_HEMOS_COLAPSADO /
# PESO_MADRID_MUERTES_TOTALES
print(f"casos_españa_estimados = {casos_españa_estimados}")
```

296.0 368.0

R0_estimada = 0.8043478260869565

casos_españa_estimados = 39600.0

1.1 Gráfico estimacion R0

Considerando solo los datos de Madrid, estimamos el R0 a partir del nº de muertos (considerando que el nº de muertos es una combinacion lineal del nº de enfermos), por lo que es posible calcular el ratio igual.

Para calcular el R0, sacamos la suma de muertos de la última semana, entre la suma de muertos de la semana anterior.

```
[8]: from datetime import datetime, timedelta
import seaborn as sns
from matplotlib import pyplot as plt
```

```

import matplotlib.dates as mdates

df = df_master

def calcular_estimaciones_R0(df):
    def calcular_R0_dia(dia,df):
        dia_semana_anterior = dia - timedelta(days=7)
        return dia,df.loc[dia:dia - timedelta(days=6)][['FALLECIDOS TOTALES_
→hoy']].sum() / df.loc[dia- timedelta(days=7):dia -_
→timedelta(days=13)][['FALLECIDOS TOTALES hoy']].sum()

    VENTANA_MEDIA_MOVIL=7

    df_R0_estimada = pd.DataFrame([calcular_R0_dia(dia,df) for dia in df.
→index[0:50]],columns=['Fecha','R0_estimada'])

    df_R0_estimada = df_R0_estimada.sort_values(by=['Fecha'], ascending=True)
    df_R0_estimada['MA_R0_estimada'] = df_R0_estimada['R0_estimada'].
→rolling(window=VENTANA_MEDIA_MOVIL).mean()
    df_R0_estimada = df_R0_estimada.sort_values(by=['Fecha'], ascending=False)
    df_R0_estimada.set_index('Fecha', inplace=True, drop=True)
    return df_R0_estimada

df= calcular_estimaciones_R0(df_master)
#df=df[['R0_estimada']]
df

chart_df=df[df.columns[-3:]]
chart_df.plot(legend=True,figsize=(13.5,9), marker='o')

plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%m-%d'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=1))
plt.xticks(rotation=45)

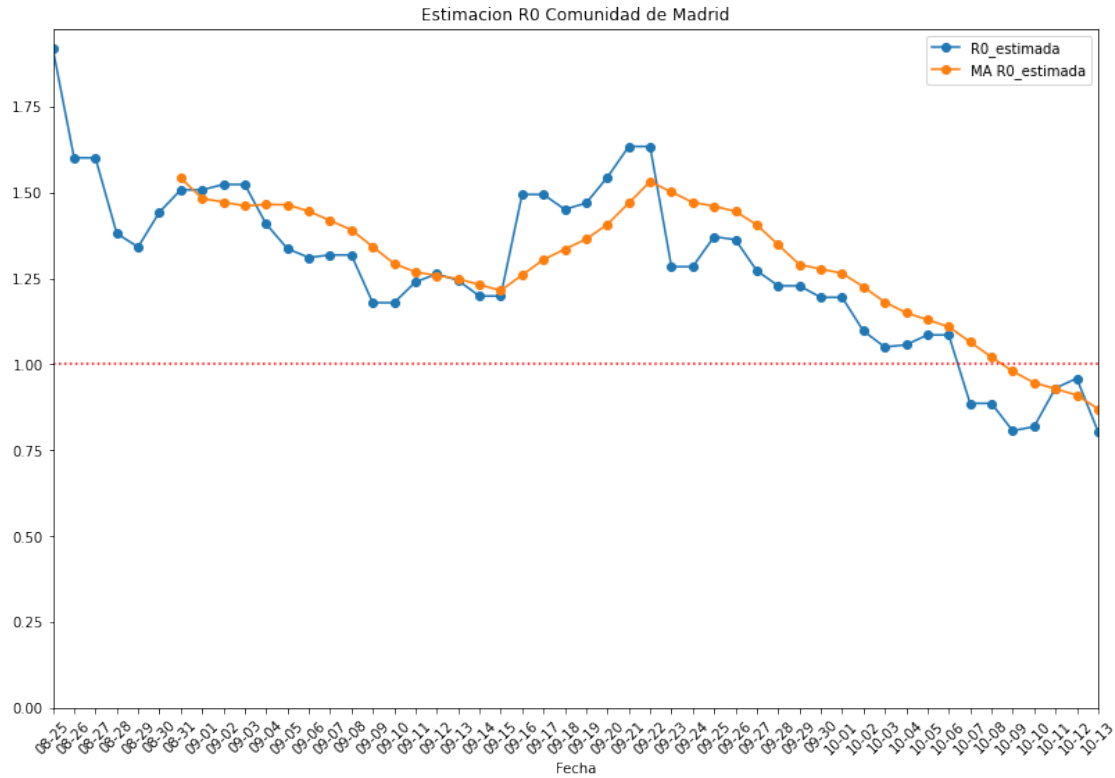
ax = plt.gca()
ax.axhline(1, color='r',linestyle = ':' )

ax.set_title("Estimacion R0 Comunidad de Madrid")
ax.set_ylim(ymin=0)

plt.show()

df.style.format ({ c : "{:20,.3f}" for c in df.columns }).
→background_gradient(cmap='Wistia', )

```



[8]: <pandas.io.formats.style.Styler at 0x7f5bc50c0978>

[9]: RO_estimada * 1.2

[9]: 0.9652173913043478

[10]: HTML("<h2>Gráfico muertes diarias en Madrid, según Comunidad de Madrid </h2>")

[10]: <IPython.core.display.HTML object>

```
[11]: import pandas as pd
import io
import matplotlib.dates as mdates
from matplotlib import pyplot as plt

df = df_master
chart_df=df[df.columns[-3:]].head(60)
chart_df.plot(legend=True,figsize=(13.5,9), marker='o')

plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%b-%d'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=7))
```

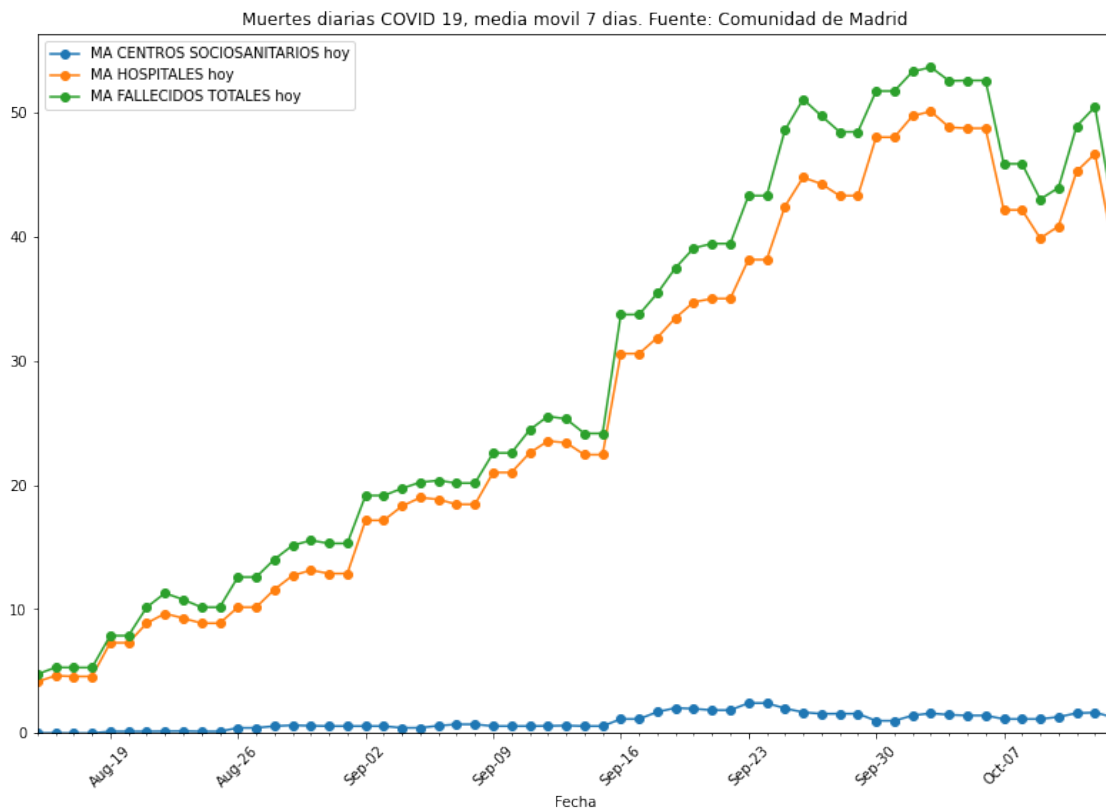


```
plt.xticks(rotation=45)

ax = plt.gca()
plt.setp(ax.get_xminorticklabels(), visible=False)

ax.set_title("Muertes diarias COVID 19, media movil_
↳"+str(VENTANA_MEDIA_MOVIL)+" dias. Fuente: Comunidad de Madrid")
ax.set_ylim(ymin=0)

plt.show()
```



```
[12]: from IPython.display import display, HTML
HTML("<h2>Comparamos los datos de hoy, de hace una semana y de un mes </h2>")
```

```
[12]: <IPython.core.display.HTML object>
```

```
[13]: from matplotlib import colors

def background_gradient(s, m, M, cmap='PuBu', low=0, high=0):
    rng = M - m
    norm = colors.Normalize(m - (rng * low),
```

```

        M + (rng * high))
    normed = norm(s.values)
    c = [colors.rgb2hex(x) for x in plt.cm.get_cmap(cmap)(normed)]
    return ['background-color: %s' % color for color in c]

df = df_master

df.style.format ({ c : "{:20,.0f}" for c in df.columns }).
    ↪background_gradient(cmap='Wistia', subset= df.columns[-3:] )

```

[13]: <pandas.io.formats.style.Styler at 0x7f5bbbed6fb38>

```

[14]: df = df_master
pd.concat([df.head(1).tail(1) , df.head(8).tail(1) , df.head(30).tail(1)]).
    ↪astype(int)[['MA HOSPITALES hoy', 'MA CENTROS SOCIO SANITARIOS hoy', 'MA_
    ↪FALLECIDOS TOTALES hoy']].style.format ({ c : "{:20,.0f}" for c in df.
    ↪columns }).background_gradient(cmap='Wistia', subset= df.columns[-3:] )

```

[14]: <pandas.io.formats.style.Styler at 0x7f5bc50b0358>

```

[15]: from IPython.display import display, HTML
HTML("<h2>Muertes medias diarias, últimos 7 días, con datos</h2>")

```

[15]: <IPython.core.display.HTML object>

```

[16]: from datetime import date

df = df_master
inicio_crisis = df.head(7).index[6]
df=df.head(7)
dia_mas_reciente = df.index[0]
dias_transcurridos_inicio_crisis = dia_mas_reciente - inicio_crisis
df = pd.DataFrame((df.head(1).max(axis=0) - df.tail(1).max(axis=0) ) /
    ↪dias_transcurridos_inicio_crisis.days ).
    ↪T[['HOSPITALES', 'DOMICILIOS', 'CENTROS SOCIO SANITARIOS', 'OTROS_
    ↪LUGARES', 'FALLECIDOS TOTALES']]
df.style.format ({ c : "{:20,.0f}" for c in df.columns }).
    ↪background_gradient(cmap='Wistia' )

```

[16]: <pandas.io.formats.style.Styler at 0x7f5bbbeddf6d8>

```

[17]: HTML("<h2>Muertes medias diarias desde que la comunidad de Madrid publica_
    ↪datos</h2>")

```

[17]: <IPython.core.display.HTML object>

```
[18]: # Calculamos los incrementos medios, desde que tenemos fechas
df = df_master
df = pd.DataFrame((df.head(1).max(axis=0) - df.tail(1).max(axis=0) ) / df.
↳shape[0] ).T[['HOSPITALES', 'DOMICILIOS', 'CENTROS SOCIO SANITARIOS', 'OTROS_
↳LUGARES', 'FALLECIDOS TOTALES']]
df.style.format ({ c : "{:20,.0f}" for c in df.columns }).
↳background_gradient(cmap='Wistia' )
```

```
[18]: <pandas.io.formats.style.Styler at 0x7f5bcb4f9908>
```

```
[ ]:
```

```
[19]: from tabula import read_pdf
from IPython.display import display, HTML
import os
import pandas as pd
import glob
import re
from tqdm.notebook import tqdm
import warnings
import os.path
fecha="201005"
import os
file_path = '../data/'+fecha+'_cam_covid19.pdf'
if not os.path.isfile(file_path):
    file_path = '../data/'+fecha+'cam_covid19.pdf'
#print("Analizando:" + file_path)
```

```
[20]: df_pdf = read_pdf(file_path,area=(300, 100, 800, 400) , pages='1')
df_pdf
```

```
Got stderr: oct 13, 2020 7:55:55 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'
```

```
[20]: [ Unnamed: 0      256.667
0      NaN      (acumulados)
1      Pacie ntes      Pacientes en
2      Hospitalizados      UCI
3      3.047      501
4      (ingresados en      (ingresados en
5      el dia)      el día)
6      59.757      5.234
7      (acumulados)      NaN
8      NaN      (acumulados)
9      NaN      NaN
```

```
10 e: Dirección General de Salud Pública, Servi... NaN]
```

```
[21]: for x,y in enumerate(df_pdf):
      print(x,"::",y)

pd.DataFrame(df_pdf)
```

```
0 :: Unnamed: 0 256.667
0 NaN (acumulados)
1 Pacie ntes Pacientes en
2 Hospitalizados UCI
3 3.047 501
4 (ingresados en (ingresados en
5 el dia) el día)
6 59.757 5.234
7 (acumulados) NaN
8 NaN (acumulados)
9 NaN NaN
10 e: Dirección General de Salud Pública, Servi... NaN
```

```
[21]: 0
0 Unn...
```

```
[22]: type(df_pdf)
```

```
[22]: list
```

```
[23]: type(df_pdf[0])
```

```
[23]: pandas.core.frame.DataFrame
```

```
[24]: total
```

```
[24]: CENTROS SOCIO SANITARIOS DOMICILIOS FALLECIDOS TOTALES \
Fecha
2020-10-13 4883 1047 17038
2020-10-09 4882 1046 16999
2020-10-08 4878 1042 16906
2020-10-07 4878 1042 16906
2020-10-06 4874 1035 16742
...
2020-04-26 4236 798 12855
2020-04-25 4170 788 12612
2020-04-24 4068 775 12252
2020-04-23 3996 769 12056
2020-04-22 3932 761 11852
```

	HOSPITALES	OTROS LUGARES	PACIENTES UCI	ACUMULADOS \
Fecha				
2020-10-13	11078	30		5445
2020-10-09	11041	30		5354
2020-10-08	10956	30		5323
2020-10-07	10956	30		5288
2020-10-06	10803	30		5262
...
2020-04-26	7800	21		0
2020-04-25	7633	21		0
2020-04-24	7388	21		0
2020-04-23	7271	20		0
2020-04-22	7144	15		0

	PACIENTES UCI DIA	HOSPITALES hoy	CENTROS SOCIO SANITARIOS hoy \
Fecha			
2020-10-13	473	37.0	1.0
2020-10-09	487	85.0	4.0
2020-10-08	487	0.0	0.0
2020-10-07	487	153.0	4.0
2020-10-06	489	0.0	0.0
...
2020-04-26	0	167.0	66.0
2020-04-25	0	245.0	102.0
2020-04-24	0	117.0	72.0
2020-04-23	0	127.0	64.0
2020-04-22	0	NaN	NaN

	FALLECIDOS TOTALES hoy
Fecha	
2020-10-13	39.0
2020-10-09	93.0
2020-10-08	0.0
2020-10-07	164.0
2020-10-06	0.0
...	...
2020-04-26	243.0
2020-04-25	360.0
2020-04-24	196.0
2020-04-23	204.0
2020-04-22	NaN

[131 rows x 10 columns]

```
[25]: get_daily_date_new_format("201005")
```

Got stderr: oct 13, 2020 7:55:56 PM

```
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'
```

```
Got stderr: oct 13, 2020 7:55:57 PM
org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
ADVERTENCIA: Using fallback font 'LiberationSans' for 'Arial,Bold'
```

[25]:

	PACIENTES UCI DIA	PACIENTES UCI ACUMULADOS	HOSPITALES DOMICILIOS	\
Fecha				
2020-10-05	501	5234	10803	1035

	CENTROS SOCIO SANITARIOS	OTROS LUGARES	FALLECIDOS	TOTALES
Fecha				
2020-10-05	4874	30		16742

[26]:

total

[26]:

	CENTROS SOCIO SANITARIOS	DOMICILIOS	FALLECIDOS	TOTALES	\
Fecha					
2020-10-13	4883	1047		17038	
2020-10-09	4882	1046		16999	
2020-10-08	4878	1042		16906	
2020-10-07	4878	1042		16906	
2020-10-06	4874	1035		16742	
...		
2020-04-26	4236	798		12855	
2020-04-25	4170	788		12612	
2020-04-24	4068	775		12252	
2020-04-23	3996	769		12056	
2020-04-22	3932	761		11852	

	HOSPITALES	OTROS LUGARES	PACIENTES UCI ACUMULADOS	\
Fecha				
2020-10-13	11078	30	5445	
2020-10-09	11041	30	5354	
2020-10-08	10956	30	5323	
2020-10-07	10956	30	5288	
2020-10-06	10803	30	5262	
...	
2020-04-26	7800	21	0	
2020-04-25	7633	21	0	
2020-04-24	7388	21	0	
2020-04-23	7271	20	0	
2020-04-22	7144	15	0	

PACIENTES UCI DIA HOSPITALES hoy CENTROS SOCIO SANITARIOS hoy \

Fecha				
2020-10-13	473	37.0		1.0
2020-10-09	487	85.0		4.0
2020-10-08	487	0.0		0.0
2020-10-07	487	153.0		4.0
2020-10-06	489	0.0		0.0
...
2020-04-26	0	167.0		66.0
2020-04-25	0	245.0		102.0
2020-04-24	0	117.0		72.0
2020-04-23	0	127.0		64.0
2020-04-22	0	NaN		NaN

FALLECIDOS TOTALES hoy

Fecha	
2020-10-13	39.0
2020-10-09	93.0
2020-10-08	0.0
2020-10-07	164.0
2020-10-06	0.0
...	...
2020-04-26	243.0
2020-04-25	360.0
2020-04-24	196.0
2020-04-23	204.0
2020-04-22	NaN

[131 rows x 10 columns]

[]: