TALLER NRO. 2, PANDAS + MATPLOTLIB

Integrantes

- Luis Febres
- Luis Jaramillo
- Renato Balcázar

```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from datetime import datetime

custom_date_parser = lambda x: datetime.strptime(x, "%Y%m%d")

# Importar ambas tablas de datos en python usando pandas. Poner la columna del índice en "MESS_DATUM" y analizar los valo
garmisch  = pd.read_csv(
    'data/garmisch.txt',
    sep=';',
    index_col=['MESS_DATUM'],
    parse_dates=['MESS_DATUM'],
    date_parser=custom_date_parser)
zugspitze = pd.read_csv(
    'data/zugspitze.txt',
    sep=';',
    index_col=['MESS_DATUM'],
    parse_dates=['MESS_DATUM'],
    date_parser=custom_date_parser)

garmisch
```

MESS_DATUM	STATIONS_ID	QN_3	FX	FM	QN_4	RSK	RSKF	SDK	SHK_TAG	NM	VPM	PM	TMK	UPM	TXK	TNK	TGK
2017-10-10	1550	10	5.6	1.0	3	0.0	6	2.300	0	7.5	9.4	937.40	9.3	82.50	15.7	4.6	3.0
2017-10-11	1550	10	-999.0	-999.0	3	0.0	0	9.533	0	2.6	9.1	938.41	9.2	81.04	18.5	2.9	1.4
2017-10-12	1550	10	-999.0	-999.0	3	0.0	0	9.483	0	1.9	9.0	941.30	10.6	75.92	22.2	3.1	1.1
2017-10-13	1550	10	-999.0	-999.0	3	0.0	0	9.483	0	3.8	10.3	944.63	10.7	82.04	19.5	4.8	2.9
2017-10-14	1550	10	-999.0	-999.0	3	0.0	0	9.617	0	0.3	9.9	944.99	11.2	78.67	21.4	3.9	1.9
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2019-04-08	1550	1	4.4	0.9	1	4.1	6	0.000	0	7.8	9.1	926.83	6.9	91.04	8.9	5.5	4.8
2019-04-09	1550	1	7.6	1.5	1	21.7	6	1.767	0	7.8	9.2	925.75	8.4	84.38	13.8	4.6	3.5
2019-04-10	1550	1	6.2	1.3	1	1.5	6	0.000	0	7.8	9.5	925.77	7.6	91.13	9.7	6.0	5.9
2019-04-11	1550	1	7.4	1.8	1	0.0	6	0.000	0	7.9	8.6	930.73	5.7	93.21	7.1	4.5	3.9
2019-04-12	1550	1	5.8	1.6	1	0.0	6	0.000	0	8.0	6.8	933.15	4.4	80.63	6.2	3.0	2.9

550 rows × 18 columns

```
# Recorta las tablas a el año 2018 [1P]
garmisch  = garmisch.loc['2018']
zugspitze = zugspitze.loc['2018']
```

```
# Volver a muestrear los datos de temperatura a promedios mensuales (" TMK") y los datos de precipitaciones a sumas mensua
garmisch_agg  = garmisch.loc[:, [' TMK', ' RSK']].resample('1m').agg({' TMK': 'mean', ' RSK': 'sum'})
garmisch_agg
```

MESS_DATUM	TMK	RSK
2018-01-31	0.806452	220.4
2018-02-28	-3.353571	55.2
2018-03-31	2.132258	50.4
2018-04-30	12.393333	34.0
2018-05-31	14.138710	114.8
2018-06-30	16.263333	152.4
2018-07-31	18.038710	99.0
2018-08-31	18.738710	199.9
2018-09-30	14.163333	118.9
2018-10-31	9.803226	65.7
2018-11-30	3.486667	11.3
2018-12-31	0.332258	166.5

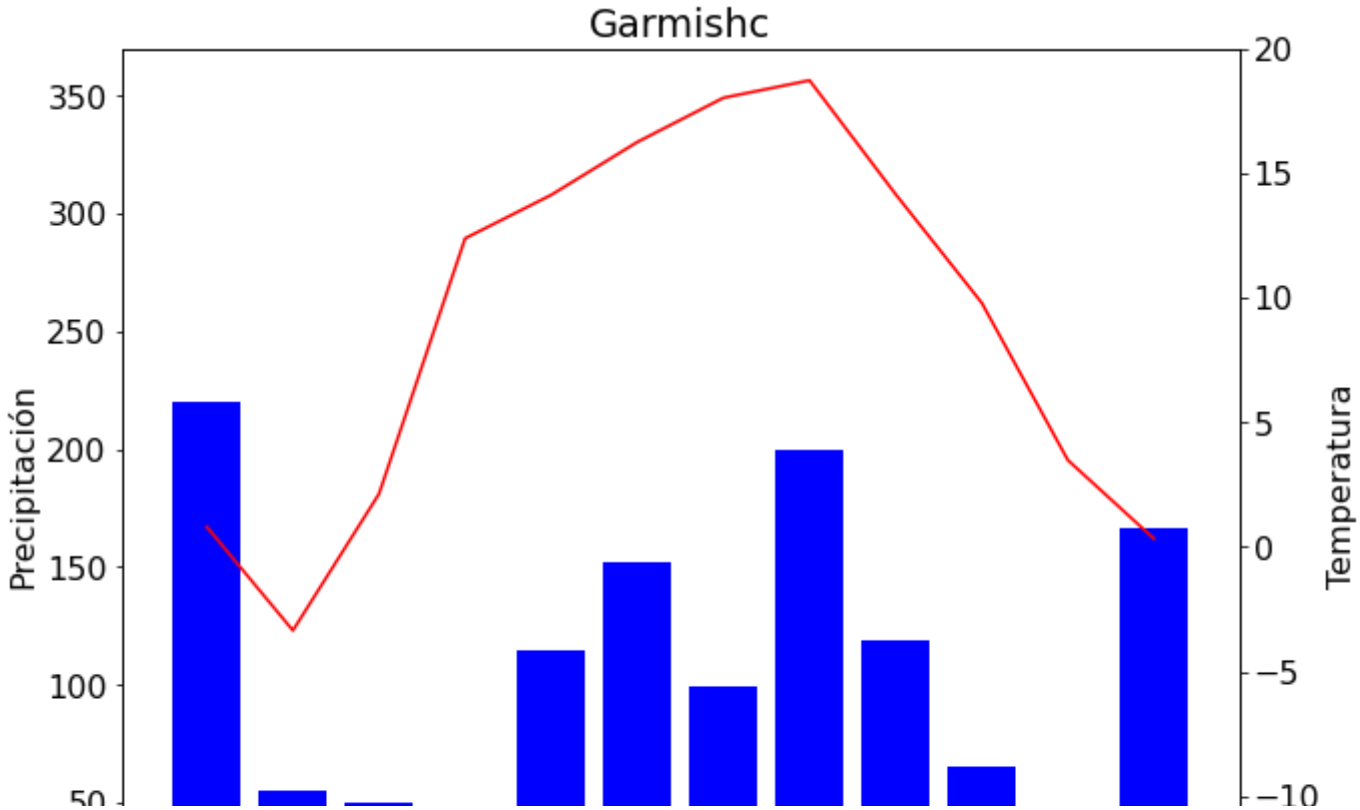
```
zugspitze_agg  = zugspitze.loc[:, [' TMK', ' RSK']].resample('1m').agg({' TMK': 'mean', ' RSK': 'sum'})
zugspitze_agg
```

MESS_DATUM	TMK	RSK
2018-01-31	-8.732258	295.6
2018-02-28	-14.764286	86.0
2018-03-31	-9.822581	148.4
2018-04-30	-1.906667	34.1
2018-05-31	0.358065	149.8
2018-06-30	2.076667	142.5

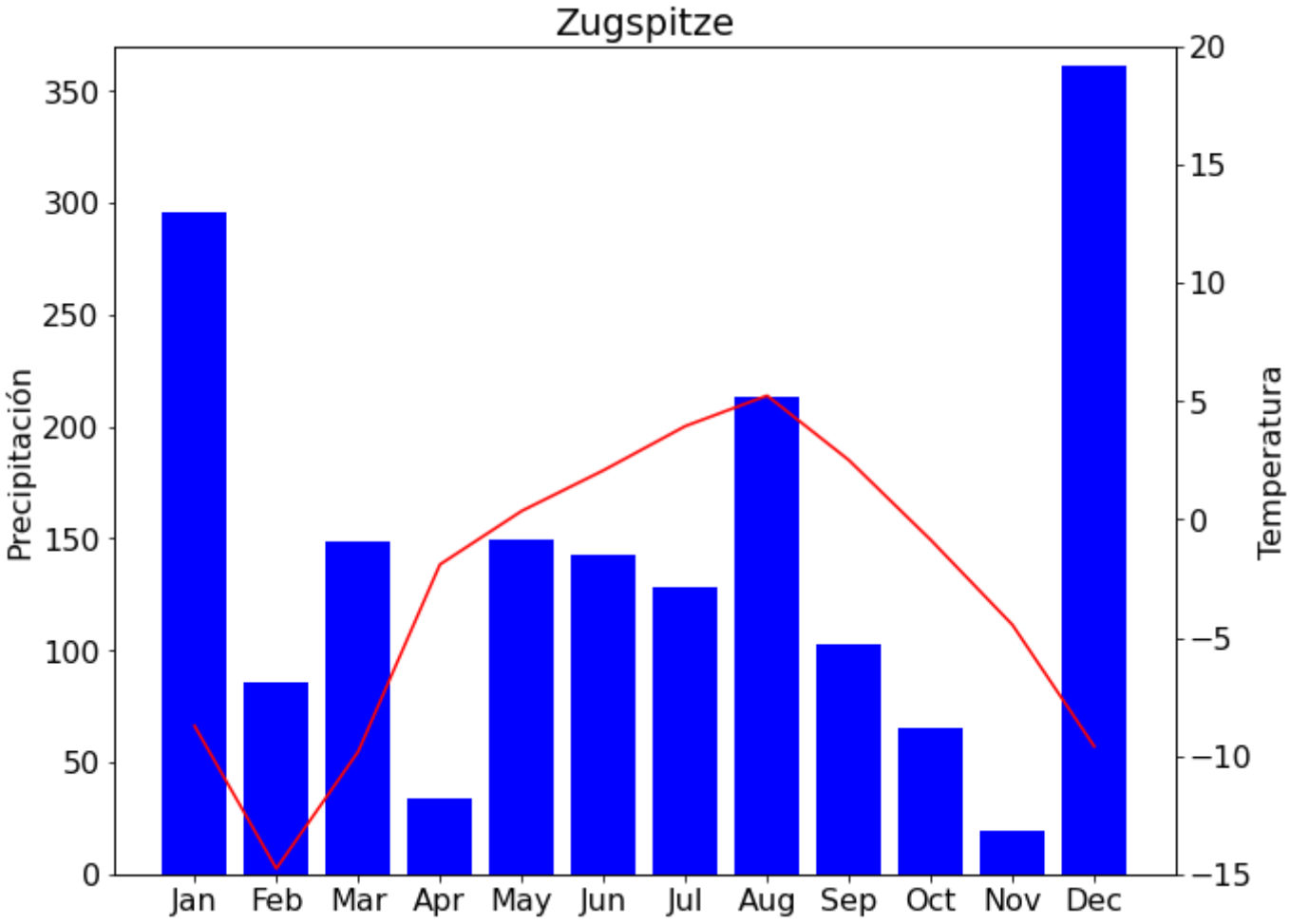
```
# Define una función de trazado que dibuja un simple diagrama climático
# Agrega los argumentos como se menciona en la lista de documentos abajo [1P]
# Establece el rango de temperatura por defecto de -15°C a 20°C y el rango de precipitaciones de 0mm a 370mm [1P]

def crear_diagrama_climatico(
        df,
        temp_col,
        prec_col,
        title,
        filename,
        temp_min=-15,
        temp_max=20,
        prec_min=0,
        prec_max=370
    ):
    """
    Dibuja un diagrama climático.

    Parametros
    ----------
    df : pd.DataFrame
        Dataframe de datos con valores para graficar
    temp_col : str
```

```
temp_col : str
    Nombre de la columna de temperatura
prec_col : str
    Nombre de la columna de precipitación
title : String
    El título para la figura
filename : String
    El nombre de la figura de salida
temp_min : Number
    El valor mínimo de temperatura a mostrar
temp_max : Number
    El valor máximo de temperatura a mostrar
prec_min : Number
    El valor mínimo de precipitación a mostrar
prec_max : Number
    El valor máximo de precipitación a mostrar

Returns
-------
La figura

"""

fig = plt.figure(figsize=(10,8))
plt.rcParams['font.size'] = 16

ax2 = fig.add_subplot(111)
ax1 = ax2.twinx()



# Dibuja los valores de temperatura como una línea roja y los valores de precipitación como barras azules: [1P]
# Pista: Revisa la documentación de matplotlib cómo trazar gráficos de barras (plt.bar?). Intenta establecer directame
# etiquetas del eje X (nombres cortos de los meses)
ax2.bar(df.index.strftime("%b"), df.loc[:,prec_col].values, color="blue")
ax1.plot(df.index.strftime("%b"), df.loc[:,temp_col].values, c="red")

# Establezca los límites apropiados para cada eje Y usando los argumentos de la función: [1P]
```

```
    ax2.set_ylim([prec_min, prec_max])
    ax1.set_ylim([temp_min, temp_max])

    # Ponga las etiquetas apropiadas a cada eje Y: [1P]
    ax2.set_ylabel("Precipitación")
    ax1.set_ylabel("Temperatura")

    # Dale a tu diagrama el título de los argumentos proporcionados: [1P]
    plt.title(title)

    # Guarda la figura como imagen png en la carpeta "output" con el nombre de archivo dado. [1P]
    #...

    final_filneame = ("output/%s.png" %(filename))
    plt.savefig(final_filneame)
    return fig




# Utilice esta función para dibujar un diagrama climático para 2018 para ambas estaciones y guarde el resultado: [1P]
crear_diagrama_climatico(df=garmisch_agg,
        temp_col=' TMK',
        prec_col=' RSK',
        title='Garmishc',
        filename='garmisch_figure',
        temp_min=-15,
        temp_max=20,
        prec_min=0,
        prec_max=370);
```




```
crear_diagrama_climatico(df=zugspitze_agg,
        temp_col=' TMK',
        prec_col=' RSK',
        title='Zugspitze',
        filename='zugspitze_figure',
        temp_min=-15,
        temp_max=20,
        prec_min=0,
        prec_max=370);
```


Zugspitze
Precipitación
Temperatura
0
50
100
150
200
250
300
350
−15
−10
−5
0
5
10
15
20
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec