

Descripción y requerimiento: Como consultor técnico de la WHO (World Health Organization) se le ha solicitado que organice la información necesaria para realizar una gráfica que: por continente, muestre la evolución del **promedio de la razón entre el número total de casos de COVID-19 y el número total de camas de hospital disponibles** a través del tiempo.

Para probar esta tarea, usted cuenta con el archivo de datos *owid-covid-data.csv* disponible para ser descargado desde la plataforma iMaster. Escriba una **función** que reciba como parámetro una cadena con la ruta dónde se encuentra guardado el archivo, incluyendo la extensión, y lo lea desde esta misma. A partir de estos datos, construya un *dataframe* sobre el cual, al utilizar el método *df.plot()* se obtenga la gráfica esperada, tal como se puede observar a continuación:

```
In [5]: df_respuesta
```

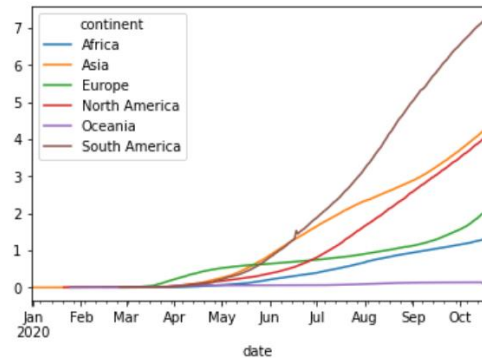
```
Out[5]:
```

	continent	Africa	Asia	Europe	North America	Oceania	South America
date							
2019-12-31		NaN	0.000004	NaN	NaN	NaN	NaN
2020-01-01		NaN	0.000004	NaN	NaN	NaN	NaN
2020-01-02		NaN	0.000004	NaN	NaN	NaN	NaN
2020-01-03		NaN	0.000007	NaN	NaN	NaN	NaN
2020-01-04		NaN	0.000007	NaN	NaN	NaN	NaN
...	
2020-10-13		1.263680	4.119695	1.901292	3.908905	0.138210	7.073163
2020-10-14		1.272543	4.156947	1.936752	3.934048	0.138368	7.109919
2020-10-15		1.283442	4.189961	1.979024	3.977127	0.138503	7.147510
2020-10-16		1.291707	4.228441	2.002530	4.012192	0.104530	7.189019
2020-10-17		1.299668	4.266616	1.872407	4.048827	0.104873	7.234603

292 rows x 6 columns

```
In [3]: df_respuesta.plot()
```

```
Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x1e819961af0>
```



Finalmente, utilice el método *df.to_dict()* para **retornar un diccionario** conteniendo la información del dataframe.

```
def caso_who(ruta_archivo_csv: str)-> dict:
    pass
```

Valide que la extensión del nombre del archivo sea de tipo csv (comma separated values). En caso contrario, **retorne** la siguiente cadena: “Extensión inválida.”. Utilice un bloque **try except** para leer el archivo. En caso de error, **retorne** la siguiente cadena: “Error al leer el archivo de datos.”

Tenga en cuenta las siguientes consideraciones:

- La fecha debe ser el **índice** principal del dataframe (**No es una columna**) y debe transformarse a un objeto tipo *fecha* de pandas. Puede realizar las siguientes comprobaciones para saber que lo está haciendo bien:

```
In [15]: df_respuesta.index.dtype
```

```
Out[15]: dtype('<M8[ns]')
```

```
In [16]: df_respuesta.index
```

```
Out[16]: DatetimeIndex(['2019-12-31', '2020-01-01', '2020-01-02', '2020-01-03',
                        '2020-01-04', '2020-01-05', '2020-01-06', '2020-01-07',
                        '2020-01-08', '2020-01-09',
                        ...,
                        '2020-10-08', '2020-10-09', '2020-10-10', '2020-10-11',
                        '2020-10-12', '2020-10-13', '2020-10-14', '2020-10-15',
                        '2020-10-16', '2020-10-17'],
                        dtype='datetime64[ns]', name='date', length=292, freq=None)
```

```
In [17]: df_respuesta.columns
```

```
Out[17]: Index(['Africa', 'Asia', 'Europe', 'North America', 'Oceania',
                'South America'],
                dtype='object', name='continent')
```

- El archivo de datos original en sus registros **NO** cuenta con la información correspondiente al total de número de casos ni de camas de hospital disponibles ni la razón entre estos. Usted debe realizar el cálculo correspondiente en función de las demás columnas que sí se encuentran en el archivo original.
- Recuerde que su función no debe imprimir la gráfica sino retornar un **diccionario**.