Algorítmica y Programación

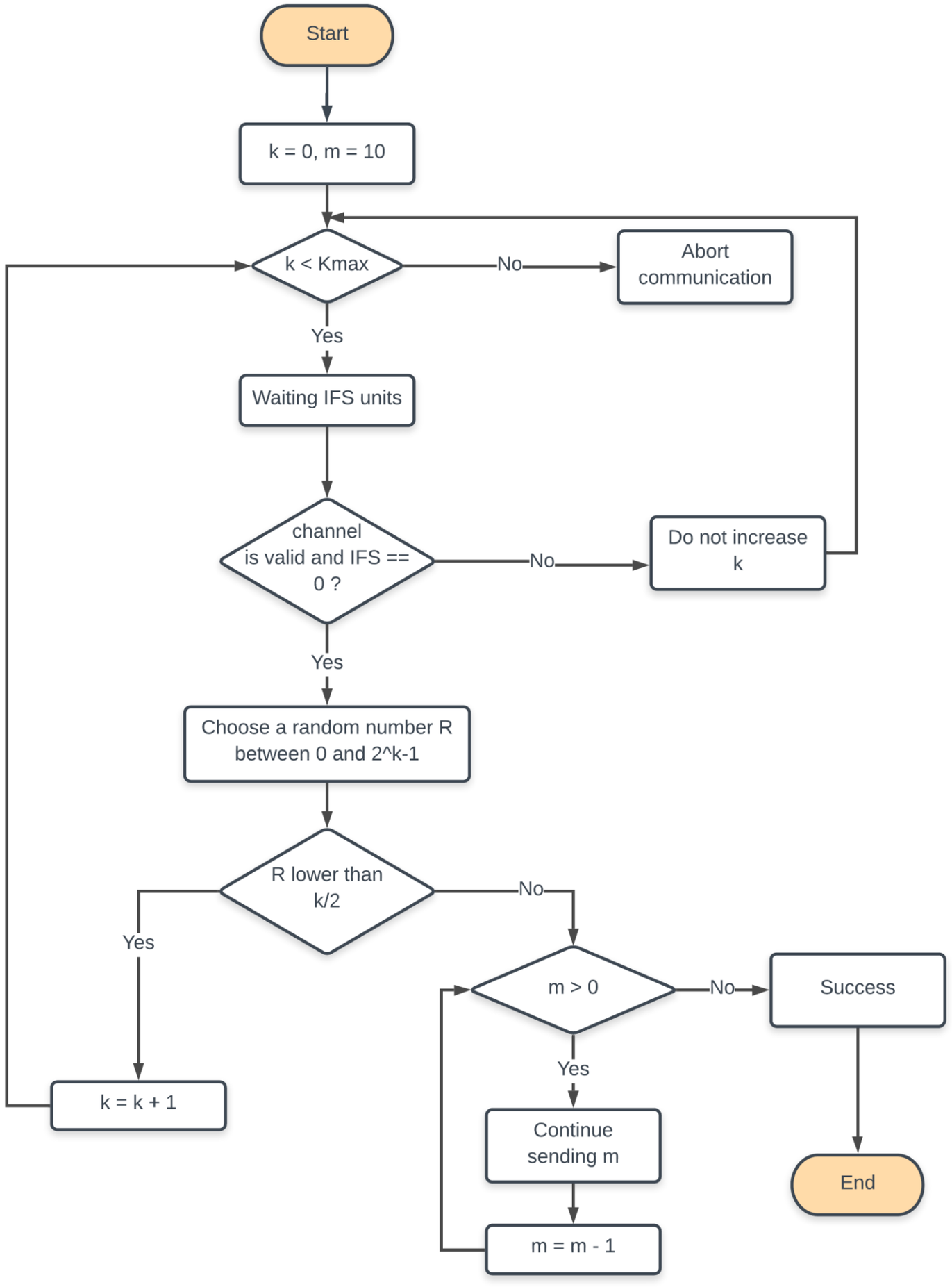
Enero - Mayo 2020

Dr. Iván S. Razo Zapata
(ivan.razo@itam.mx)

ITAM

Funciones

Funciones


Start
k = 0, m = 10
k < Kmax
No
Abort communication
Yes
Waiting IFS units
channel is valid and IFS == 0 ?
No
Do not increase k
Yes
Choose a random number R between 0 and 2^k-1
R lower than k/2
No
Yes
k = k + 1
m > 0
No
Success
Yes
Continue sending m
End
m = m - 1

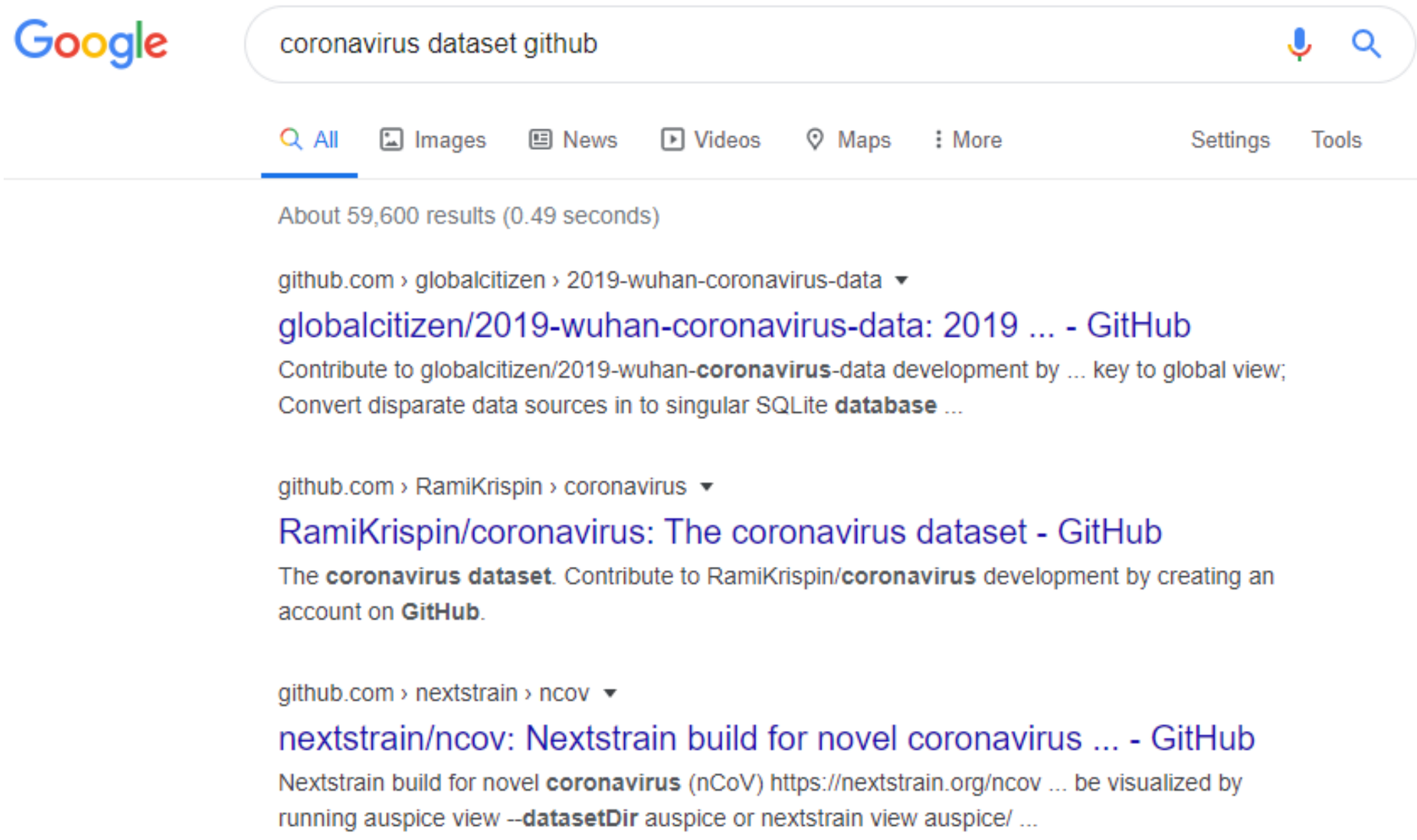
Funciones

```python
import numpy as np

k= 0
m = 10
kmax = 20
IFS = 15
channel = True

while k < kmax:
    print("Waiting IFS units")
    # Decrease IFS
    if (channel == True) and (IFS == 0):
        R = np.random.uniform(low=0, high=2**(k-1), size=1)
        print("R = ", R)
        if R < (k/2):
            k = k + 1
        else:
            while m < 0:
                print("Sending m ")
                m -= 1
            print("Successful transmission")
            break
    else:
        print("Do not increase k")
        # channel = True or False
```

Funciones

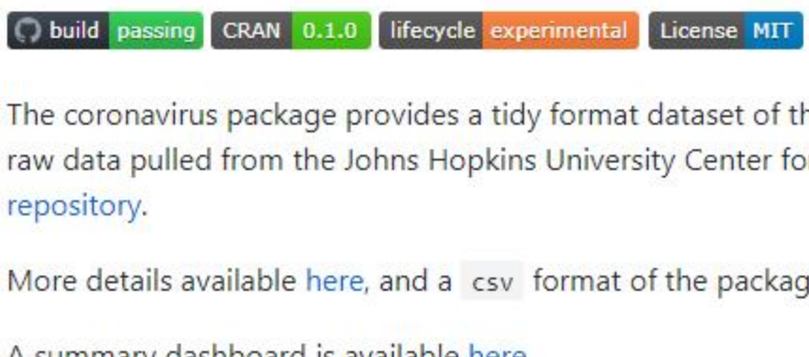
- Crear funciones básicas
  - Suma3
  - Multiplica3
- Importar y llamar desde otro archivo

Funciones


Google
coronavirus dataset github
All
Images
News
Videos
Maps
More
Settings
Tools
About 59,600 results (0.49 seconds)
github.com › globalcitizen › 2019-wuhan-coronavirus-data
globalcitizen/2019-wuhan-coronavirus-data: 2019 ... - GitHub
Contribute to globalcitizen/2019-wuhan-coronavirus-data development by ... key to global view; Convert disparate data sources in to singular SQLite database ...
github.com › RamiKrispin › coronavirus
RamiKrispin/coronavirus: The coronavirus dataset - GitHub
The coronavirus dataset. Contribute to RamiKrispin/coronavirus development by creating an account on GitHub.
github.com › nextstrain › ncov
nextstrain/ncov: Nextstrain build for novel coronavirus ... - GitHub
Nextstrain build for novel coronavirus (nCoV) https://nextstrain.org/ncov ... be visualized by running auspice view --datasetDir auspice or nextstrain view auspice/ ...

Funciones

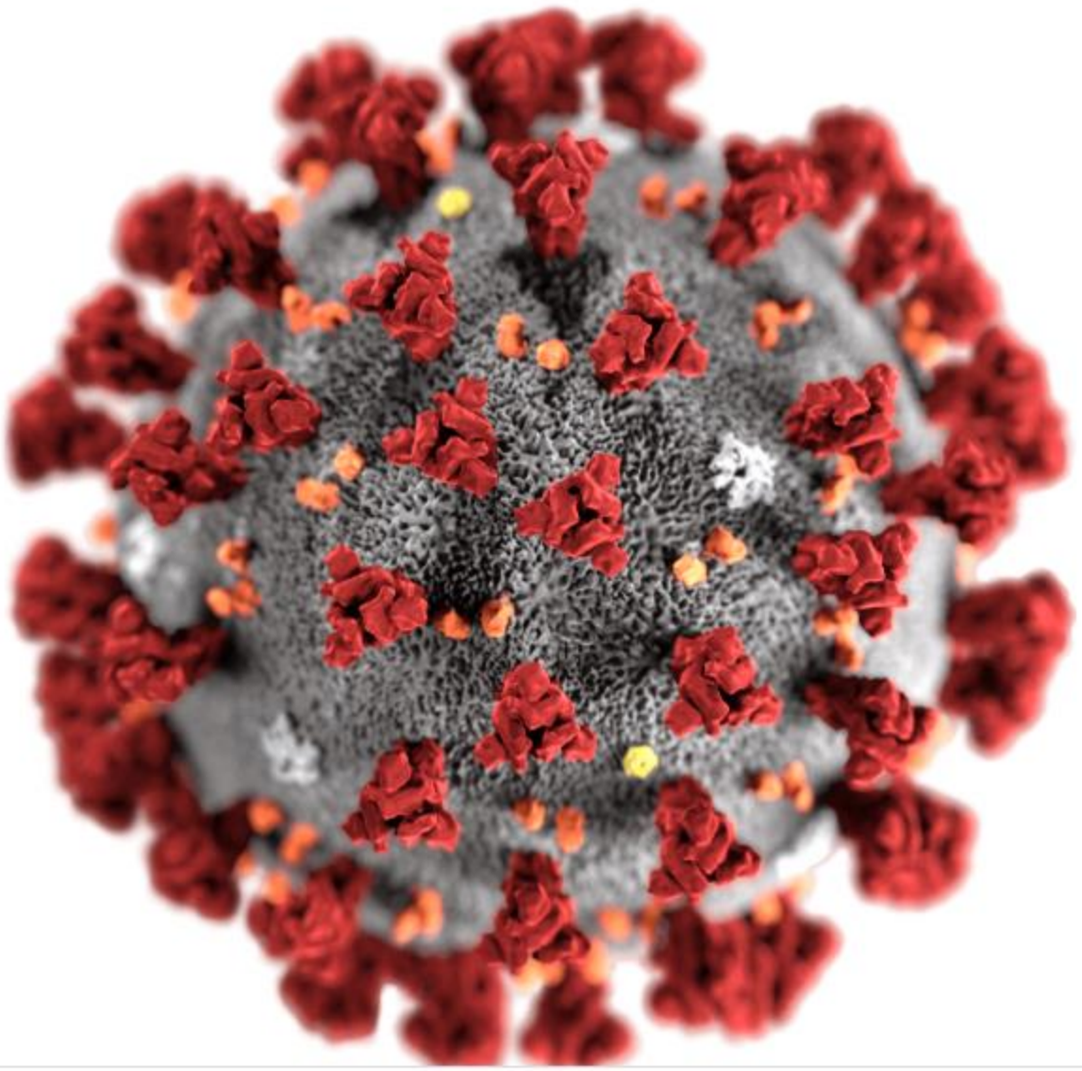
coronavirus



The coronavirus package provides a tidy format dataset of the 2019 Novel Coronavirus COVID-19 (2019-nCoV) epidemic. The raw data pulled from the Johns Hopkins University Center for Systems Science and Engineering (JHU CCSE) Coronavirus repository.

More details available here, and a `csv` format of the package dataset available here

A summary dashboard is available here



Funciones

CSV format for the coronavirus R package dataset

23 commits | 1 branch | 0 packages | 0 releases | 1 contributor

Branch: master | New pull request | Find file | Clone or download

RamiKrispin push updates — Latest commit 8f1c912 15 hours ago

coronavirus_dataset.csv	push updates	15 hours ago
coronavirus_dataset.xlsx	push updates	15 hours ago

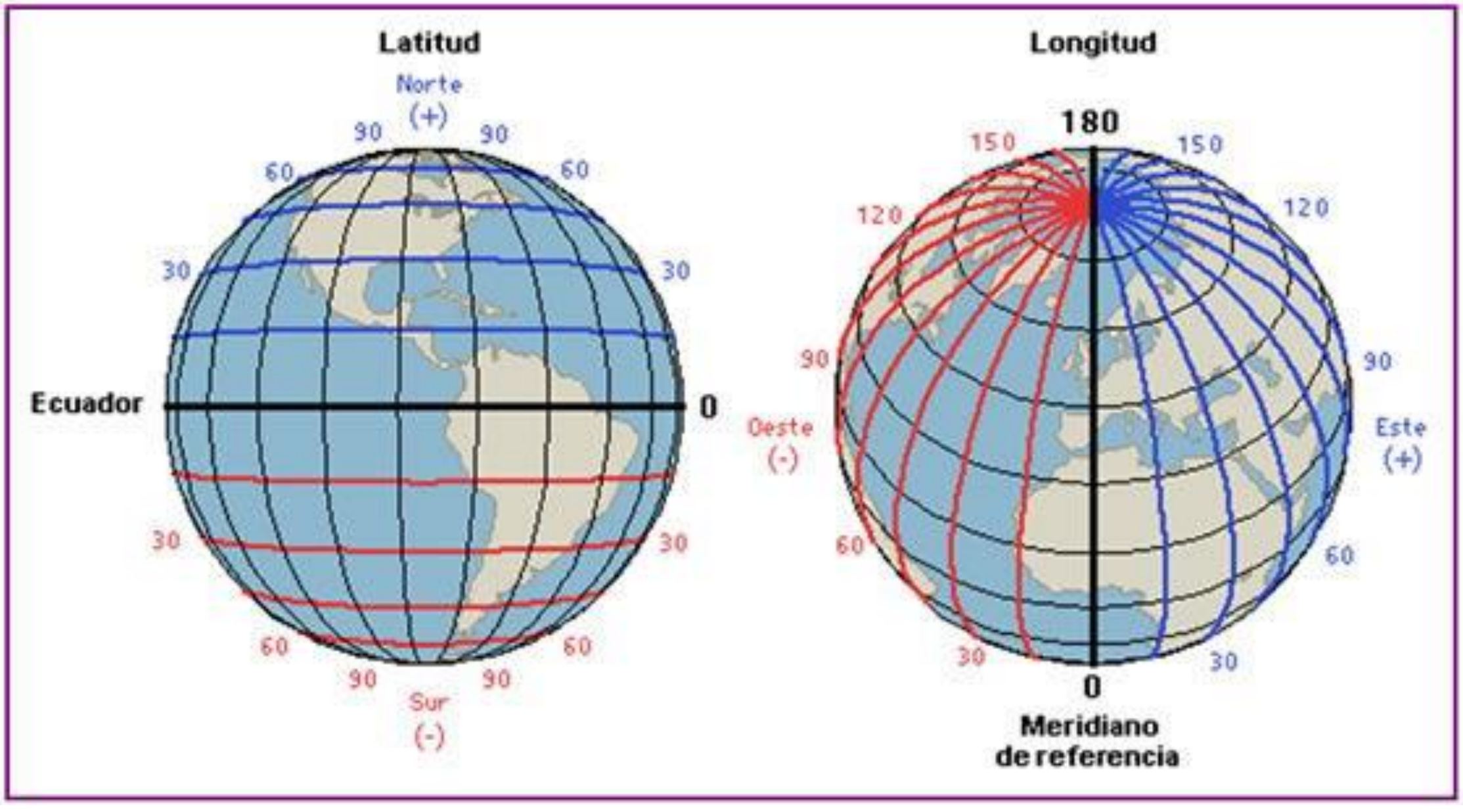
Funciones

- Crear DataFrame
- Index Country.Region
- Examinar por país
- Crear una lista
- Crear una serie

Funciones

- Crear funciones
  - Agregado por país
  - Agregado por “zona/región”
  - Agregado por fechas
  - Agregado por tipo y país

Funciones


Latitud
Norte
(+)
90
90
60
60
30
30
Ecuador
0
30
30
60
60
90
90
Sur
(-)
Longitud
180
150
150
120
120
90
90
Oeste
(-)
Este
(+)
60
60
30
30
0
Meridiano
de referencia

Funciones

```python
def agregadoPais(dataFrame, pais):
    dataF = dataFrame.loc[pais,['cases','type']].groupby('type').sum()
    return dataF

def agregadoZona(dataFrame,lat1,lat2,long1,long2):
    dataF = dataFrame[(dataFrame['Lat'] >= lat1) & \
                      (dataFrame['Lat'] <= lat2) & \
                      (dataFrame['Long'] >= long1) & \
                      (dataFrame['Long'] <= long2)]
    return dataF

def agregadoFecha(dataFrame,fecha1,fecha2):
    dataF = dataFrame[(dataFrame['date'] >= fecha1) & (dataFrame['date'] <= fecha2)]
    return dataF

def agregadoTipoPais(dataFrame,Tipo,Pais):
    dataF = dataFrame.loc[((dataFrame.type == Tipo) &  \
                           (dataFrame.index == Pais)), \
        ['date','cases']].groupby('date').agg(['sum']).cumsum()
    return dataF
```

Funciones

```
import pandas as pd
import funciones01 as f1

dataF1 = pd.read_csv('DataSets/coronavirus-csv-master/coronavirus_dataset.csv')
dataF1.set_index('Country.Region', inplace=True)

pais = 'Mexico'
newDataF = f1.agregadoPais(dataF1,pais)
print(pais)
print(newDataF)

pais = 'Japan'
newDataF = f1.agregadoPais(dataF1,pais)
print(pais)
print(newDataF)

pais = 'US'
newDataF = f1.agregadoPais(dataF1,pais)
print(pais)
print(newDataF)

newDataF = f1.agregadoZona(dataF1,14,30,-110,-88)
print(newDataF)

newDataF = f1.agregadoZona(dataF1,14,80,-110,-88)
print(newDataF)

```

Funciones

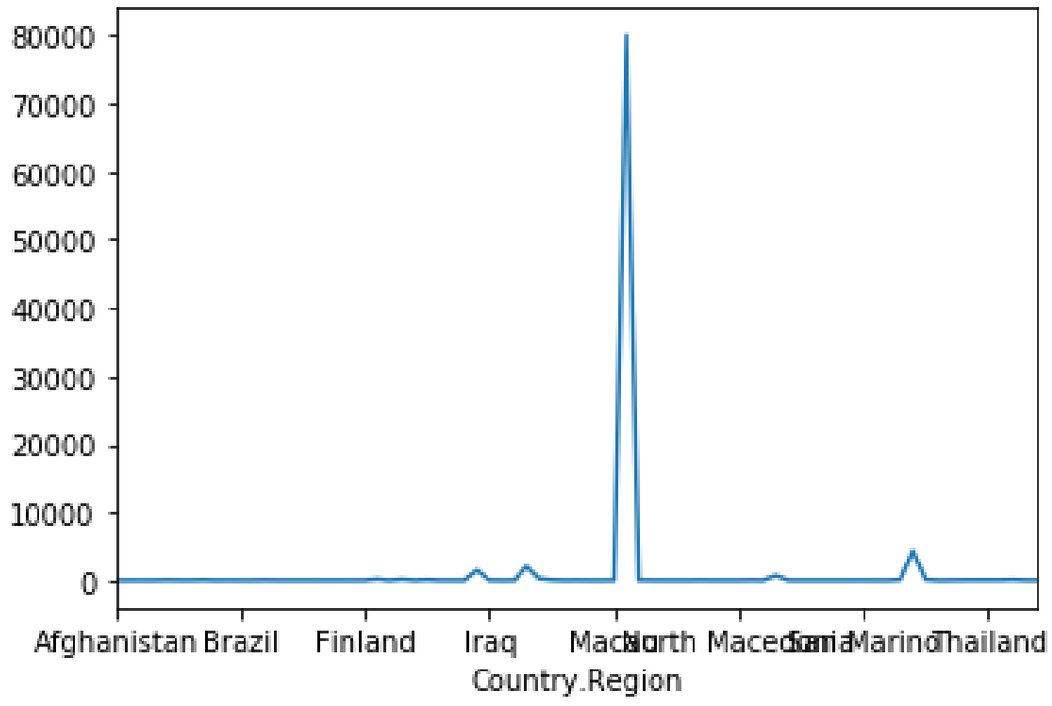
```
35 newDataF = f1.agregadoFecha(dataF1,'2020-01-01','2020-01-05')
36 print(newDataF.cases.sum())
37
38 newDataF = f1.agregadoFecha(dataF1,'2020-01-05','2020-02-28')
39 print(newDataF.cases.sum())
```

Funciones

```
58
59 newDataF = f1.agregadoTipoPais(dataF1,'confirmed', 'Canada')
60 print(newDataF)
61
62 newDataF2 = f1.agregadoTipoPais(dataF1,'death', 'Canada')
63 print(newDataF2)
64
65 newDataF3 = f1.agregadoTipoPais(dataF1,'confirmed',pais)
66 print(newDataF3)
67
68 newDataF4 = f1.agregadoTipoPais(dataF1,'death',pais)
69 print(newDataF4)
70
```

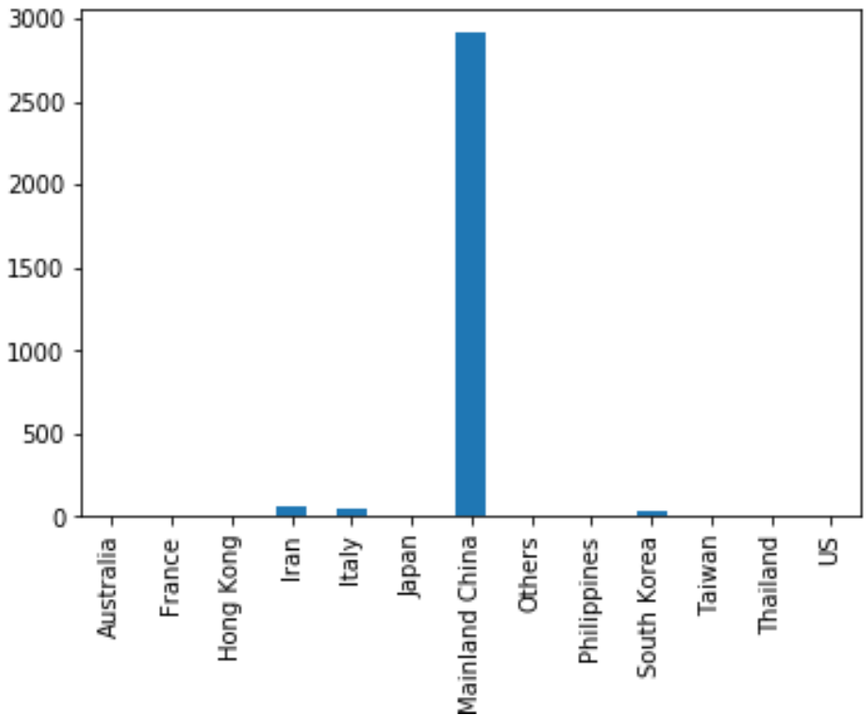
Plots

```
In [14]: dataF1.loc[(dataF1.type == "confirmed")].groupby('Country.Region').sum().cases.plot()
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x1f36a9d18d0>
```



Plots

```
In [15]: dataF1.loc[(dataF1.type == "death")].groupby('Country.Region').sum().cases.plot(x='Country.Region', \
    ...:             y='cases', kind="bar")
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x1f36aa40b70>
```



Plots

```
In [16]: dataF1.loc[((dataF1.type == "confirmed") & (dataF1.index == 'Mexico')),['date', \
    ...:                'cases']].groupby('date').agg(['sum']).cumsum().plot(y='cases', kind='bar')
Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x1f36a913e80>
```