

Aplicando Python a datos políticos

INDICE

1. Carga de data frame con pandas (no estudiar)
2. Listas
3. Diccionarios
4. Estructuras de control
5. Funciones

=====

1. Carga de data frame con pandas (no estudiar)

```
In [1]: import pandas as pd  
import numpy as np
```

```
In [2]: # pip install openpyxl
```

```
In [145... aprobacion= pd.read_excel("aprobacion.xlsx")
```

```
In [146... aprobacion.head()
```

```
Out[146...
```

	pais	anio	trimestre	presidente	presidente_genero	aprobacion_neta	pib	corr
0	Argentina	2000	1	Fernando de la Rúa	Masculino	40	14	552151
1	Argentina	2000	2	Fernando de la Rúa	Masculino	16	14	552151
2	Argentina	2000	3	Fernando de la Rúa	Masculino	24	14	552151
3	Argentina	2000	4	Fernando de la Rúa	Masculino	-18	14	552151
4	Argentina	2001	1	Fernando de la Rúa	Masculino	-7	14	527807

Hay numeros que se visualizan en notacion cientifica por ser muy largos. vamos a corregirlo

```
In [100... pd.set_option('display.float_format', lambda x: f'{x:.0f}')
```

```
In [101... aprobacion.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1020 entries, 0 to 1019
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   pais                  1020 non-null  object
1   anio                  1020 non-null  int64
2   trimestre             1020 non-null  int64
3   presidente            1020 non-null  object
4   presidente_genero     1020 non-null  object
5   aprobacion_neta       1020 non-null  float64
6   pib                   1020 non-null  int64
7   corrupcion            1020 non-null  float64
8   poblacion             1020 non-null  int64
9   desempleo            1020 non-null  float64
10  crecimiento_pib       1020 non-null  float64
dtypes: float64(4), int64(4), object(3)
memory usage: 87.8+ KB
```

```
In [102... aprobacion.describe()
```

```
Out[102...      anio  trimestre  aprobacion_neta      pib      corrupcion  poblacion  des
count  1020         1020              1020      1020          1020          1020
mean    2007          2              15  415993735215580  411402368992  31379665
std       4          1              28  258665895292915  687789664895  47560715
min    2000          1             -66   642001953125   17373017702  3030347
25%    2003          2              -7  212949919700623   40130146629  5867626
50%    2007          2              16  375663681030273   91972026250  13072060
75%    2011          3              38  662462463378906  400986433732  30317848
max    2014          4              87  944239273071289  3139626109943  204213133
```

2.Listas

Pueden repasar los metodos de creación de listas en el material de computación. Nosotros vamos a crarlas rapidamente desde el dataframe para trabajar sobre ellas

```
In [103... # Convertir cada columna a una lista
pais_list = aprobacion['pais'].tolist()
anio_list = aprobacion['anio'].tolist()
trimestre_list = aprobacion['trimestre'].tolist()
presidente_list = aprobacion['presidente'].tolist()
presidente_genero_list = aprobacion['presidente_genero'].tolist()
aprobacion_neta_list = aprobacion['aprobacion_neta'].tolist()
pib_list = aprobacion['pib'].tolist()
corrupcion_list = aprobacion['corrupcion'].tolist()
poblacion_list = aprobacion['poblacion'].tolist()
desempleo_list = aprobacion['desempleo'].tolist()
crecimiento_pib_list = aprobacion['crecimiento_pib'].tolist()
```

```
In [104... presidente_list[0:5]
```

```
Out[104... ['Fernando de la Rúa',
             'Fernando de la Rúa',
             'Fernando de la Rúa',
             'Fernando de la Rúa',
             'Fernando de la Rúa']
```

```
In [105... pais_unique = list(set(pais_list))
anio_unique = list(set(anio_list))
trimestre_unique = list(set(trimestre_list))
presidente_unique = list(set(presidente_list))
presidente_genero_unique = list(set(presidente_genero_list))
aprobacion_neta_unique = list(set(aprobacion_neta_list))
pib_unique = list(set(pib_list))
corrupcion_unique = list(set(corrupcion_list))
poblacion_unique = list(set(poblacion_list))
desempleo_unique = list(set(desempleo_list))
crecimiento_pib_unique = list(set(crecimiento_pib_list))
```

```
In [106... presidente_unique[:11]
```

```
Out[106... ['Jorge Quiroga Ramírez',
            'Porfirio Lobo Sosa',
            'Eduardo Alberto Duhalde',
            'Juan Orlando Hernández',
            'Vicente Fox',
            'Alan García',
            'Gonzalo Sánchez de Lozada',
            'Michelle Bachelet',
            'Valentín Paniagua',
            'Alberto Fujimori',
            'Evo Morales']
```

Metodos para listas

```
In [107... # 1. Longitud de la Lista
print(len(pais_unique)) # No modifica la lista original
```

In [108...

```
# 2. Ordenar La Lista
print(pais_unique)

pais_unique.sort() # Modifica La lista original [ascendete]
print(pais_unique)

pais_unique.sort(reverse=True) # Modifica La lista original [descendente]
print(pais_unique)

['Brasil', 'Costa Rica', 'Nicaragua', 'Uruguay', 'Perú', 'Venezuela', 'Argentina',
'Chile', 'Ecuador', 'México', 'Panamá', 'Colombia', 'Paraguay', 'El Salvador', 'Boli
via', 'Honduras', 'Guatemala']
['Argentina', 'Bolivia', 'Brasil', 'Chile', 'Colombia', 'Costa Rica', 'Ecuador', 'El
Salvador', 'Guatemala', 'Honduras', 'México', 'Nicaragua', 'Panamá', 'Paraguay', 'Pe
rú', 'Uruguay', 'Venezuela']
['Venezuela', 'Uruguay', 'Perú', 'Paraguay', 'Panamá', 'Nicaragua', 'México', 'Hondur
as', 'Guatemala', 'El Salvador', 'Ecuador', 'Costa Rica', 'Colombia', 'Chile', 'Bra
sil', 'Bolivia', 'Argentina']
```

In [109...

```
# 3. Invertir el Orden de La Lista
pais_unique.reverse() # Modifica La lista original
print(pais_unique)

['Argentina', 'Bolivia', 'Brasil', 'Chile', 'Colombia', 'Costa Rica', 'Ecuador', 'El
Salvador', 'Guatemala', 'Honduras', 'México', 'Nicaragua', 'Panamá', 'Paraguay', 'Pe
rú', 'Uruguay', 'Venezuela']
```

In [110...

```
# 4. Agregar Elementos a La Lista
pais_unique.append('Jamaica') # Modifica La lista original
print(pais_unique)

pais_unique.insert(1, 'EEUU') # Modifica La lista original
print(pais_unique)

['Argentina', 'Bolivia', 'Brasil', 'Chile', 'Colombia', 'Costa Rica', 'Ecuador', 'El
Salvador', 'Guatemala', 'Honduras', 'México', 'Nicaragua', 'Panamá', 'Paraguay', 'Pe
rú', 'Uruguay', 'Venezuela', 'Jamaica']
['Argentina', 'EEUU', 'Bolivia', 'Brasil', 'Chile', 'Colombia', 'Costa Rica', 'Ecuad
or', 'El Salvador', 'Guatemala', 'Honduras', 'México', 'Nicaragua', 'Panamá', 'Parag
uay', 'Perú', 'Uruguay', 'Venezuela', 'Jamaica']
```

In [111...

```
# 5. Eliminar Elementos de La Lista
pais_unique.remove('Chile') # Modifica La lista original
print(pais_unique)

del pais_unique[2] # Modifica La lista original
print(pais_unique)

last_element = pais_unique.pop() # Modifica La lista original y guarda el elemento
print(pais_unique)
print(f'Elemento eliminado: {last_element}')
```

```
['Argentina', 'EEUU', 'Bolivia', 'Brasil', 'Colombia', 'Costa Rica', 'Ecuador', 'El Salvador', 'Guatemala', 'Honduras', 'México', 'Nicaragua', 'Panamá', 'Paraguay', 'Perú', 'Uruguay', 'Venezuela', 'Jamaica']
['Argentina', 'EEUU', 'Brasil', 'Colombia', 'Costa Rica', 'Ecuador', 'El Salvador', 'Guatemala', 'Honduras', 'México', 'Nicaragua', 'Panamá', 'Paraguay', 'Perú', 'Uruguay', 'Venezuela', 'Jamaica']
['Argentina', 'EEUU', 'Brasil', 'Colombia', 'Costa Rica', 'Ecuador', 'El Salvador', 'Guatemala', 'Honduras', 'México', 'Nicaragua', 'Panamá', 'Paraguay', 'Perú', 'Uruguay', 'Venezuela']
Elemento eliminado: Jamaica
```

```
In [112... # 6. Buscar un Elemento en La Lista
print('Argentina' in pais_unico) # No modifica la lista original

index = pais_unico.index('Argentina') # No modifica la lista original
print(f'Índice de Argentina: {index}')
```

True
Índice de Argentina: 0

```
In [113... # 7. Contar Elementos en La Lista
count = pais_unico.count('Argentina') # No modifica la lista original
print(f'Cantidad de veces que aparece Argentina: {count}')
```

Cantidad de veces que aparece Argentina: 1

```
In [114... # 8. Unir Listas
anio_unico = [2020, 2021, 2022, 2023, 2024]
combined_list = pais_unico + anio_unico # Crea una nueva lista combinada
print(combined_list)
```

```
['Argentina', 'EEUU', 'Brasil', 'Colombia', 'Costa Rica', 'Ecuador', 'El Salvador', 'Guatemala', 'Honduras', 'México', 'Nicaragua', 'Panamá', 'Paraguay', 'Perú', 'Uruguay', 'Venezuela', 2020, 2021, 2022, 2023, 2024]
```

```
In [115... # 10. Convertir la Lista a una Cadena
pais_string = ', '.join(pais_unico) # Crea una nueva cadena
print(pais_string)
type(pais_string) # comprobamos el tipo de dato
```

Argentina, EEUU, Brasil, Colombia, Costa Rica, Ecuador, El Salvador, Guatemala, Honduras, México, Nicaragua, Panamá, Paraguay, Perú, Uruguay, Venezuela

Out[115... str

```
In [116... # 10. Obtener un subconjunto de la lista
primeros_tres = pais_unico[:3]
primeros_tres
```

Out[116... ['Argentina', 'EEUU', 'Brasil']

algunos metodos mas pero para datos numericos

```
In [117... desempleo_unico = [5.2, 7.1, 6.5, 4.8, 5.9, 8.3, 7.0, 5.4]

# 1. Obtener el valor mínimo y máximo
min_desempleo = min(desempleo_unico)
```

```
print(min_desempleo)
max_desempleo = max(desempleo_unique)
print(max_desempleo)
```

4.8
8.3

```
In [118... # 2. Suma de todos los elementos
suma_desempleo = sum(desempleo_unique)
suma_desempleo
```

Out[118... 50.2

```
In [119... # 3. Promedio de los elementos
promedio_desempleo = sum(desempleo_unique) / len(desempleo_unique)
print(f"Promedio: {promedio_desempleo}")
```

Promedio: 6.275

```
In [120... # Otra forma de calcular promedio

import statistics
```

```
In [121... statistics.mean(desempleo_unique)
```

Out[121... 6.275

```
In [122... # 4. Redondear los valores
desempleo_redondeado = [round(num, 0) for num in desempleo_unique]
desempleo_redondeado
```

Out[122... [5.0, 7.0, 6.0, 5.0, 6.0, 8.0, 7.0, 5.0]

Diccionario a partir de listas

Nos apuramos y quisimos generar rapidamente un diccionario con nuestras listas_unique pero hubo un problema...

```
In [123... # Crear un diccionario con 5 elementos
aprobacion_dict = {
    'pais': pais_unique[:5], # Primeros 5 elementos de la columna 'pais'
    'anio': anio_unique[:5], # Primeros 5 elementos de la columna 'anio'
    'presidente': presidente_unique[:5], # Primeros 5 elementos de la columna 'pre
    'aprobacion_neta': aprobacion_neta_unique[:5], # Primeros 5 elementos de la co
    'pib': pib_unique[:5] # Primeros 5 elementos de la columna 'pib'
}
aprobacion_dict
```

```
Out[123...] {'pais': ['Argentina', 'EEUU', 'Brasil', 'Colombia', 'Costa Rica'],
'anio': [2020, 2021, 2022, 2023, 2024],
'presidente': ['Jorge Quiroga Ramírez',
'Porfirio Lobo Sosa',
'Eduardo Alberto Duhalde',
'Juan Orlando Hernández',
'Vicente Fox'],
'aprobacion_neta': [1.12699997425079,
2.927000046,
3.76200008392334,
4.059999943,
5.1300001139999996],
'pib': [666188659667969,
497184448242188,
278633213043213,
291961498260498,
702841491699219]}
```

Estos valores nunca van a tener sentido porque con unique rompimos la relacion original de los datos entre sí.

Diccionario a partir de listas pero con datos manuales

```
In [124...] # SE UTILIZAN NUMEROS AL AZAR SOLO PARA EJEMPLIFICAR
sample_dict = {
'pais': ['Argentina', 'Brasil', 'Chile', 'Uruguay', 'Colombia'],
'presidente': ['Javier Milei', 'Lula Da Silva', 'Gabriel Boric ', 'Luis Lacalle
'aprobacion_neta': [50.5, 55.0, 35.2, 34.5, 41.0],
'pib': [100000, 300000, 90000, 30000, 60000],
'poblacion': [46000000, 210000000, 19000000, 3500000, 51000000],
'desempleo': [13.5, 3.0, 7.2, 6.1, 7.5],
'crecimiento_pib': [-2.5, 2.5, 1, 0.8, 0.5]
}
```

```
In [125...] sample_dict
```

```
Out[125...] {'pais': ['Argentina', 'Brasil', 'Chile', 'Uruguay', 'Colombia'],
'presidente': ['Javier Milei',
'Lula Da Silva',
'Gabriel Boric ',
'Luis Lacalle Pou',
'Gustavo Petro'],
'aprobacion_neta': [50.5, 55.0, 35.2, 34.5, 41.0],
'pib': [100000, 300000, 90000, 30000, 60000],
'poblacion': [46000000, 210000000, 19000000, 3500000, 51000000],
'desempleo': [13.5, 3.0, 7.2, 6.1, 7.5],
'crecimiento_pib': [-2.5, 2.5, 1, 0.8, 0.5]}
```

```
In [126...] # Métodos básicos de diccionarios
print(sample_dict.keys()) # Muestra las claves
```

```
dict_keys(['pais', 'presidente', 'aprobacion_neta', 'pib', 'poblacion', 'desempleo',
'crecimiento_pib'])
```

```
In [127... print(sample_dict.values()) # Muestra los valores

dict_values(['Argentina', 'Brasil', 'Chile', 'Uruguay', 'Colombia'], ['Javier Milei', 'Lula Da Silva', 'Gabriel Boric ', 'Luis Lacalle Pou', 'Gustavo Petro'], [50.5, 55.0, 35.2, 34.5, 41.0], [100000, 300000, 90000, 30000, 60000], [46000000, 21000000, 19000000, 3500000, 51000000], [13.5, 3.0, 7.2, 6.1, 7.5], [-2.5, 2.5, 1, 0.8, 0.5]])

In [128... print(sample_dict.items()) # Muestra las pares clave-valor

dict_items([('pais', ['Argentina', 'Brasil', 'Chile', 'Uruguay', 'Colombia']), ('presidente', ['Javier Milei', 'Lula Da Silva', 'Gabriel Boric ', 'Luis Lacalle Pou', 'Gustavo Petro']), ('aprobacion_neta', [50.5, 55.0, 35.2, 34.5, 41.0]), ('pib', [100000, 300000, 90000, 30000, 60000]), ('poblacion', [46000000, 21000000, 19000000, 3500000, 51000000]), ('desempleo', [13.5, 3.0, 7.2, 6.1, 7.5]), ('crecimiento_pib', [-2.5, 2.5, 1, 0.8, 0.5])])

In [129... print(sample_dict.get("poblacion")) # Obtiene el valor asociado a una clave

[46000000, 21000000, 19000000, 3500000, 51000000]

In [130... sample_dict["ideología"] = ["globalista",None,"globalista","globalista","nacionalis

In [131... sample_dict

Out[131... {'pais': ['Argentina', 'Brasil', 'Chile', 'Uruguay', 'Colombia'],
'presidente': ['Javier Milei',
'Lula Da Silva',
'Gabriel Boric ',
'Luis Lacalle Pou',
'Gustavo Petro'],
'aprobacion_neta': [50.5, 55.0, 35.2, 34.5, 41.0],
'pib': [100000, 300000, 90000, 30000, 60000],
'poblacion': [46000000, 21000000, 19000000, 3500000, 51000000],
'desempleo': [13.5, 3.0, 7.2, 6.1, 7.5],
'crecimiento_pib': [-2.5, 2.5, 1, 0.8, 0.5],
'ideología': ['globalista', None, 'globalista', 'globalista', 'nacionalista']}
```

4.Estructuras de control

for

```
In [132... # Convertir cada elemento de la lista redondeada en enteros
desempleo_redondeado
desempleo_entero = [int(num) for num in desempleo_redondeado]
print(desempleo_entero)

[5, 7, 6, 5, 6, 8, 7, 5]

Una variante del ejemplo anterior

In [133... # Lista vacía para almacenar los valores enteros
desempleo_entero2 = []

# Iterar sobre los elementos de la lista redondeada y convertir a enteros
```



```

for num in desempleo_redondeado:
    desempleo_entero2.append(int(num))

# Mostrar la Lista resultante
print(desempleo_entero2)

```

[5, 7, 6, 5, 6, 8, 7, 5]

5.Funciones

In [134...

aprobacion

Out[134...

	pais	anio	trimestre	presidente	presidente_genero	aprobacion_neta	
0	Argentina	2000	1	Fernando de la Rúa	Masculino	40	1401525
1	Argentina	2000	2	Fernando de la Rúa	Masculino	16	1401525
2	Argentina	2000	3	Fernando de la Rúa	Masculino	24	1401525
3	Argentina	2000	4	Fernando de la Rúa	Masculino	-18	1401525
4	Argentina	2001	1	Fernando de la Rúa	Masculino	-7	1401525
...
1015	Venezuela	2013	4	Nicolás Maduro	Masculino	-17	9442392
1016	Venezuela	2014	1	Nicolás Maduro	Masculino	-18	9442392
1017	Venezuela	2014	2	Nicolás Maduro	Masculino	-19	9442392
1018	Venezuela	2014	3	Nicolás Maduro	Masculino	-22	9442392
1019	Venezuela	2014	4	Nicolás Maduro	Masculino	-25	9442392

1020 rows × 11 columns

In [142...

```

# Creamos una funcion que nos devuelva si el crecimiento del PBI es
# negativo o positivo para el pais y años indicados:

def crecimiento_PBI(pais_buscado, ano_buscado):
    for i in range(len(pais_list)):
        if pais_list[i] == pais_buscado and ano_list[i] == ano_buscado:
            crecimiento = crecimiento_pib_list[i]
            if crecimiento > 0:
                return 'Positivo'

```

```

        elif crecimiento < 0:
            return 'Negativo'
        else:
            return 'Neutro'

    return f"No se encontró el crecimiento del PIB para {pais_buscado} en el año {a

# Ejemplo de uso:
resultado = crecimiento_PBI('Venezuela', 2013)
print(resultado)
resultado = crecimiento_PBI('Venezuela', 2014)
print(resultado)

```

Positivo

Negativo

In [136... crecimiento_PBI('Argentina', 2014)

Out[136... 'Negativo'

In [137... *# Ahora queremos crear una funcion que nos devuelva el calculo de
PBI per capita (PBI / poblacion) para el pais y año indicados*

```

def pbi_pc(pais_buscado, ano_buscado):
    for i in range(len(pais_list)):
        if pais_list[i] == pais_buscado and ano_list[i] == ano_buscado:
            pib = pib_list[i]
            poblacion = poblacion_list[i]
            return pib / poblacion

    return f"No se encontró el PIB o la población para {pais_buscado} en el año {an

```

In [140... pbi_pc('Argentina', 2000)

Out[140... 3782033.6560151516